



TESIS - KS142501

EKSTRAKSI DAN KONVERSI *EVENT LOGS* DARI  
BASIS DATA SISTEM RUMAH SAKIT DAN *OPEN*  
*SOURCE* ERP DENGAN ANTARMUKA BAHASA  
ALAMI YANG TERSTRUKTUR UNTUK KEBUTUHAN  
*PROCESS MINING*

NAMBI SEMBILU

NRP. 05211450010004

DOSEN PEMBIMBING

Dr.Eng. FEBRILIYAN SAMOPA, S.Kom., M.Kom

MAHENDRAWATI E.R., S.T., M.Sc., Ph.D

PROGRAM MAGISTER

JURUSAN SISTEM INFORMASI

FAKULTAS TEKNOLOGI INFORMASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2018



THESIS - KS142501

EXTRACTION AND CONVERSION OF EVENT LOGS  
FROM THE DATABASE OF THE HOSPITAL SYSTEM  
AND OPEN SOURCE ERP WITH A STRUCTURED  
NATURAL LANGUAGE INTERFACE FOR THE  
NEEDS OF THE MINING PROCESS

NAMBI SEMBILU

NRP. 05211450010004

SUPERVISORS

Dr.Eng., FEBRILIYAN SAMOPA, S.Kom., M.Kom

MAHENDRAWATI E.R., S.T., M.Sc., Ph.D

MASTER PROGRAM

INFORMATION SYSTEM

FACULTY OF INFORMATION TECHNOLOGY

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2018

*Halaman ini sengaja dikosongkan*

## LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
Magister Komputer (M.Kom)  
di  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
Oleh:

Nambi Sembilu  
NRP. 05211450010004

Tanggal Ujian : 5 Juni 2018  
Periode Wisuda : September 2018

Disetujui Oleh:

Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom  
NIP. 19650310 19910 2 1001

(Pembimbing)

Mahendrawathi E. R., S.T., M.Sc., Ph.D  
NIP. 19761011 20060 4 2001

(Pembimbing)

Prof. Ir. Arif Djunaidy, M.Sc., Ph.D  
NIP. 19581005 19860 3 1003

(Penguji)

Erma Suryani, S.T., M.T. Ph.D  
NIP. 19700427 20050 1 2001

(Penguji)



Dekan

Fakultas Teknologi Informasi dan Komunikasi

Dr.H.Agus Zainal Arifin, S.Kom., M.Kom  
NIP. 19720809 19951 2 1001



*Halaman ini sengaja dikosongkan*

**EKSTRAKSI DAN KONVERSI EVENT LOGS DARI BASIS DATA  
SISTEM RUMAH SAKIT DAN *OPEN SOURCE* ERP DENGAN  
ANTARMUKA BAHASA ALAMI YANG TERSTRUKTUR UNTUK  
KEBUTUHAN *PROCESS MINING***

Nama Mahasiswa : Nambi Sembilu

NRP : 05211450010004

Dosen Pembimbing: 1. Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom  
2. Mahendrawati ER., S.T., M.Sc., Ph.D

**ABSTRAK**

Process mining merupakan sebuah cara untuk mengetahui ataupun memperbaiki alur proses sistem yang berjalan di sebuah organisasi. Process mining dapat digunakan dalam berbagai macam bidang serta dilakukan oleh siapapun yang memiliki keahlian dalam sebuah bidang. Untuk dapat melakukan process mining dibutuhkan sumberdata yang disebut dengan catatan kejadian (*event log*). Untuk mengambil sumber informasi yang tersimpan dalam basis data diperlukan bahasa komputer yang disebut SQL (*Strucured Query Language*). Sering kali seseorang yang ahli dalam bidang tertentu (Kesehatan, Sipil, Bisnis, Perbankan, dll) tidak memiliki pengetahuan lebih tentang bahasa komputer. Sehingga menyulitkan para ahli bidang tersebut untuk mengambil informasi yang mereka butuhkan dari sebuah basis data. Dengan kata lain untuk mengambil event log yang dibutuhkan dalam process mining harus melibatkan Ahli IT. Melibatkan Ahli IT pastinya membutuhkan waktu dan biaya yang tidak sedikit, belum lagi jika Ahli IT tersebut tidak sepenuhnya memahami process mining dan event log.

Oleh karena itu untuk meringankan permasalahan tersebut diperlukan sebuah media perantara antara ahli bidang dengan basis data agar memudahkan pengambilan informasi (dalam hal ini Event Log). Salah satu cara mengambil informasi dari basis data tanpa menggunakan bahasa SQL adalah dengan NLIDB (Natural Language Interface to A Database), yaitu dengan memanfaatkan bahasa alami yang sering digunakan manusia untuk komunikasi sebagai masukan dengan keluaran hasil informasi yang dibutuhkan sesuai keinginan.

Dari hasil pengujian pada penelitian ini telah berhasil membuat sebuah alat ekstraksi *event log* dengan NLIDB dengan nilai akurasi sebesar 84%. Dengan begitu alat yang dihasilkan pada penelitian dapat digunakan untuk keperluan ekstraksi *event log* dalam bidang proses mining. Dengan menggunakan konsep pemetaan informasi semi otomatis seorang analis bisnis dapat melihat semua *event log* secara langsung yang berjalan dalam sebuah sistem dengan menuliskan perintah bahasa alami kemudian akan ditampilkan dalam bentuk tabel.

**Kata kunci:** proses mining, ekstraksi event log, *natural language interface to a database*.





*Halaman ini sengaja dikosongkan*

**EVENT LOGS EXTRACTION AND CONVERSION FROM  
DATABASE OF MEDICAL SYSTEM AND *OPEN SOURCE* ERP  
USING STRUCTURED NATURAL LANGUAGES INTERFACE  
FOR PROCESS MINING NEEDS**

Student Name : Nambi Sembilu

Student Number : 05211450010004

Supervisor : 1. Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom  
2. Mahendrawati ER., S.T., M.Sc., Ph.D

**ABSTRACT**

Process mining is a way of knowing or repair process flow system running on an organization. Process mining can be used in a variety of fields as well as conducted by anyone with expertise in a field. To be able to do the process mining required a data source called the note events (event log). To retrieve the information stored in the databases needed computer language called SQL (Strucured Query Language). Often times someone who are experts in specific areas (health, Civic, business, banking, etc.) have no more knowledge of the language of the computer. So difficult experts the field to retrieve the information they need from a database. In other words, to take the required event log in process mining should involve the IT Experts. Involving Experts IT certainly takes time and cost is not a little, not to mention if IT Experts do not fully understand the process mining and the event log.

Therefore, to alleviate these problems required a medium between the experts of the field with the database in order to facilitate the retrieval of information (in this case the Event Log). One way to retrieve information from the database without using the SQL language is by NLIDB (Natural Language Interface to A Database), that is, by making use of natural language that is often used for human communication as entered by the output of the results information needed as you wish.

From the results of testing on this research have successfully created a tool event log extraction with NLIDB with a value of 84% accuracy. So the resulting tool in research can be used for the purposes of extraction of the event log in the field of process mining. By using the concept of mapping information semi-automatic business analyst can see all event log directly that runs in a system with write natural language commands will then be shown in the form of a table.

**Keyword:** *process mining, event logs extraction, natural language interface to a database.*



*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas berkat rahmat dan ridho-Nya sehingga penulis dapat menyelesaikan tesis dengan judul “EKSTRAKSI DAN KONVERSI EVENT LOGS PADA BASIS DATA SISTEM RUMAH SAKIT DAN ODOO ERP DENGAN MENGGUNAKAN ANTARMUKA BAHASA ALAMI YANG TERSTRUKTUR UNTUK KEBUTUHAN PROCESS MINING”. Penyusunan tesis ini dibuat sebagai salah satu syarat kelulusan program magister jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Penulis menyadari selama menempuh pendidikan dan proses penyelesaian tesis ini penulis memperoleh bantuan dan dukungan dari berbagai pihak. Dalam kesempatan kali ini, penulis mengucapkan terima kasih yang sebesar-besarnya kepada pihak-pihak yang membantu pengerjaan tesis ini, antara lain:

1. Kedua orang tua, kedua kakak dan keluarga yang selalu mendukung dan memotivasi penulis selama ini
2. Bapak Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom dan Ibu Mahendrawathi E. R., S.T., M.Sc., Ph.D yang selalu sabar dan telaten dalam membimbing penulis dalam menyelesaikan tesis ini
3. Bapak Prof. Ir. Arif Djunaidy, M.sc., Ph.D dan Erma Suryani, S.T., M.T. Ph.D. yang telah memberikan banyak kritik dan saran untuk perbaikan penelitian ini.
4. Seluruh Bapak dan Ibu dosen serta karyawan di program magister jurusan Sistem Informasi ITS yang telah membagikan ilmu dan inspirasi kepada penulis
5. Calon istri tercinta Astri Suryaningsih yang telah memberikan dukungan, motivasi dan kasih sayang yang tulus kepada penulis.
6. Rekan-rekan keluarga besar program magister Sistem Informasi ITS angkatan 2014 yang telah memberikan bantuan dan dukungan

kepada penulis selama mengikuti perkuliahan dan proses penelitian ini berlangsung.

7. Teman-teman dan pihak lain yang tidak dapat penulis cantumkan namanya satu per satu yang telah mendoakan, memberikan bantuan, dukungan serta sumbangan pemikiran dalam proses penyelesaian tesis ini.

Semoga Allah SWT senantiasa memberikan berkat dan anugerah-Nya serta membalas semua kebaikan yang telah dilakukan. Penulis menyadari banyak kekurangan yang terdapat dalam penelitian ini, oleh karena itu kritik dan saran yang bersifat membangun akan selalu diterima oleh penulis. Semoga penelitian ini dapat memberikan manfaat dan wawasan yang berguna bagi pengembangan ilmu pengetahuan dan bagi pembaca.

Surabaya, 26 Maret 2018

Penulis

*Halaman ini sengaja dikosongkan*





## DAFTAR ISI

LEMBAR PENGESAHAN .....	iii
ABSTRAK .....	v
ABSTRACT .....	vii
KATA PENGANTAR .....	ix
DAFTAR ISI .....	xii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xvii
BAB I PENDAHULUAN .....	1
1.1. Latar belakang .....	1
1.2. Perumusan masalah .....	4
1.3. Batasan masalah .....	4
1.4. Tujuan penelitian .....	5
1.5. Lingkup penelitian .....	5
1.6. Kontribusi penelitian .....	5
1.7. Sistematika penulisan .....	6
BAB II KAJIAN PUSTAKA .....	7
2.1. Penelitian Terdahulu .....	7
2.2. Proses Bisnis .....	11
2.3. <i>Process mining</i> .....	13
2.4. <i>Event log</i> .....	16
2.4.1. <i>Event logs</i> pada sistem informasi .....	18
2.5. <i>Natural Language Processing (NLP)</i> .....	20
2.5.1. Bidang pengetahuan dalam NLP .....	21
2.6. <i>Natural Language Interface to a Database (NLIDB)</i> .....	22
2.6.1. Sub Komponen NLIDB .....	23
2.6.2. Tahapan NLIDB .....	24
2.6.3. Keuntungan NLIDB .....	25
2.7. <i>Receiver Operating Characteristic (ROC)</i> .....	27
BAB III METODE PENELITIAN .....	32

3.1. Metode Penelitian.....	30
3.2. Penyiapan data.....	30
3.3. Pembuatan bahasa alami yang terstruktur .....	31
3.4. Mendefinisikan informasi <i>event logs</i> dalam basis data .....	32
3.5. Pemrosesan bahasa alami ke dalam bahasa SQL .....	34
3.5.1. <i>Scanner (Tokenizing)</i> .....	34
3.5.2. Analisis sintak ( <i>Parser</i> ) .....	35
3.5.3. Analisis semantic ( <i>Translator</i> ) .....	35
3.5.4. Analisis pragmatik ( <i>Evaluator</i> ) .....	36
3.6. Perancangan desain aplikasi ekstraksi <i>event log</i> .....	36
3.6.1. Rancangan konversi bahasa ke dalam format XES .....	39
3.7. Pengujian alat ekstraksi .....	40
3.8. Analisis hasil .....	41
BAB IV PEMBAHASAN .....	43
4.1. Penyiapan data.....	43
4.1.1. Pengambilan data.....	43
4.1.2. Pengumpulan data.....	43
4.1.3. Penentuan proses bisnis dan aktivitas.....	44
4.2. Pembuatan bahasa alami yang terstruktur .....	45
4.2.1. Unsur bahasa alami atribut event log .....	48
4.2.2. Kalimat perintah ekstraksi.....	49
4.2.3. Struktur kalimat perintah ekstraksi.....	51
4.3. Mendefinisikan informasi <i>event logs</i> dalam basis data .....	54
4.3.1. Sumber tabel dan kolom .....	55
4.4. Pemrosesan bahasa alami ke dalam bahasa SQL .....	61
4.4.1. <i>Scanner (Tokenizing)</i> .....	61
4.4.2. Analisis sintak ( <i>Parser</i> ) .....	61
4.4.2.1. Konversi unsur waktu bahasa alami.....	66
4.4.3. Analisis semantik ( <i>Translator</i> ) .....	66
4.4.4. Analisis pragmatik ( <i>Evaluaoir</i> ) .....	79
4.5. Perancangan alat ekstraksi dengan bahasa alami .....	79
4.5.1. Perancangan Basis data .....	79

4.5.1.1. Identifikasi Entitas .....	80
4.5.1.2. Identifikasi Relasi .....	82
4.5.1.3. Identifikasi Atribut dan entitas .....	83
4.5.2. Perancangan Antarmuka .....	87
4.6. Uji coba alat ekstraksi .....	89
4.6.1. Uji coba ekstraksi dengan bahasa alami .....	90
4.6.1.1. Pengujian sesuai KPE .....	90
4.6.1.2. Pengujian tidak sesuai KPE .....	93
4.6.2. Uji coba konversi format XES .....	98
4.6.3. Uji coba <i>process mining</i> dengan aplikasi ProM .....	100
4.6.4. Uji akurasi ekstraksi event logs dengan bahasa alami .....	102
4.6.5. <i>User Acceptance Test</i> (UAT) oleh analis bisnis .....	107
4.7. Analisa hasil .....	108
BAB V KESIMPULAN DAN SARAN .....	120
5.1 Kesimpulan .....	120
5.2 Saran .....	121
DAFTAR PUSTAKA .....	123
LAMPIRAN A .....	127
LAMPIRAN B .....	129
LAMPIRAN C .....	131
LAMPIRAN D .....	133
LAMPIRAN E .....	137
LAMPIRAN F .....	139
BIOGRAFI PENULIS .....	140

*Halaman ini sengaja dikosongkan*



## DAFTAR GAMBAR

Gambar 2.1	Tiga Jenis Utama dari Process Mining .....	17
Gambar 2.2	Tiga Tipe Dasar Process mining.....	18
Gambar 2.3	Struktur Umum Event Log .....	23
Gambar 2.4	Kriteria ROC.....	30
Gambar 3.1	Alur Metode Penelitian.....	33
Gambar 3.2	Pemetaan Ekstraksi event logs dalam database.....	36
Gambar 3.3	Konsep desain aplikasi Evel .....	33
Gambar 3.4	Alur Proses Ekstrask pada aplikasi Evel .....	40
Gambar 3.5	Use Case pada aplikasi Evel .....	40
Gambar 4.1	Contoh Bentuk relasi atribut <i>event logs</i> dalam desain basis data ..	55
Gambar 4.2	Bentuk relasi tabel atribut <i>event logs</i> pada aktivitas Permintaan barang, persetujuan permintaan dan penerimaan barang .....	57
Gambar 4.3	Bentuk <i>message log</i> pada sistem Odoo ERP .....	58
Gambar 4.4	Bentuk relasi sumber event logs pada Odoo ERP .....	59
Gambar 4.5	Form pengaturan akses aplikasi.....	89
Gambar 4.6	Form pembuatan akses basis data.....	89
Gambar 4.7	Form pemetaan informasi event log .....	90
Gambar 4.8	Form ekstraksi <i>event logs</i> dengan bahasa alami .....	90
Gambar 4.9	Contoh salah satu hasil pengujian untuk KPE.....	94
Gambar 4.10	Contoh pengujian tidak sesuai kaidah KPE.....	95
Gambar 4.11	Contoh pengujian atribut <i>event log</i> .....	96
Gambar 4.12	Contoh pengujian operator tanggal.....	98
Gambar 4.13	Contoh operator bulan .....	99
Gambar 4.14	Contoh pengujian operator tahun .....	99
Gambar 4.15	Konfigurasi atribut Event logs dengan masukan dataCSV ...	100
Gambar 4.16	Hasil konversi dengan <i>tools</i> satcos.....	100
Gambar 4.17	Hasil pengujian format XES aplikasi Disco .....	101
Gambar 4.18	Hasil pengujian heuristic miner aplikasi ProM .....	102
Gambar 4.19	Hasil pengujian inductive miner aplikasi ProM .....	102

Gambar 4.20	kurva ROC untuk ekstraksi event log .....	104
Gambar 4.21	Contoh bentuk fitur otomatis saran .....	104
Gambar 4.22	kurva ROC algoritma php similar_text .....	106
Gambar 4.23	kurva ROC algoritma LCS .....	106
Gambar 4.24	Contoh bentuk fitur otomatis perbaikan.....	106
Gambar 4.25	Bentuk isi tabel farmasi_permintaan.....	108
Gambar 4.26	Bentuk isi tabel farmasi_permintaan_unit .....	109
Gambar 4.27	Bentuk isi tabel pengguna .....	109
Gambar 4.28	Contoh hasil ekstraksi sistem Rumah Sakit dengan SQL.....	110
Gambar 4.29	Contoh hasil ekstraksi sistem Rumah Sakit dengan NLIDB ..	110
Gambar 4.30	Bentuk isi tabel mail_message .....	112
Gambar 4.31	Bentuk isi tabel mail_tracking_value.....	112
Gambar 4.32	Bentuk isi tabel res_partner.....	113
Gambar 4.33	Bentuk isi tabel res_users.....	113
Gambar 4.34	Contoh hasil ekstraksi sistem Odoo ERP dengan SQL .....	114
Gambar 4.35	Contoh hasil ekstraksi sistem Odoo ERP dengan NLIDB.....	114
Gambar 4.36	Gambaran proses ekstraksi dan konversi event logs .....	115





*Halaman ini sengaja dikosongkan*

## DAFTAR TABEL

Tabel 2.1	Contoh Event Log.....	20
Tabel 2.2	Kontingensi ROC .....	30
Tabel 4.1	Spesifikasi Sistem yang akan dijadikan ujicoba.....	45
Tabel 4.2	Struktur kalimat perintah .....	48
Tabel 4.3	Isi atribut event logs dalam bahasa alami .....	49
Tabel 4.4	Daftar kata yang diabaikan .....	63
Tabel 4.5	Daftar kata pembantu.....	63
Tabel 4.6	Daftar atribut dan operator .....	63
Tabel 4.7	Daftar frasa yang diproses .....	66
Tabel 4.8	Identifikasi Entitas aplikasi <i>Evenl</i> .....	78
Tabel 4.9	Relasi antar entitas aplikasi <i>Evenl</i> .....	83
Tabel 4.10	Atribut untuk tabel users.....	84
Tabel 4.11	Atribut untuk tabel roles .....	84
Tabel 4.12	Atribut untuk tabel event logs client.....	84
Tabel 4.13	Atribut untuk tabel event logs process .....	85
Tabel 4.14	Atribut untuk event logs activity .....	85
Tabel 4.15	Atribut untuk tabel event logs source tables.....	85
Tabel 4.16	Atribut untuk tabel event logs source columns .....	86
Tabel 4.17	Atribut untuk tabel event logs source parameters .....	86
Tabel 4.18	Atribut untuk tabel source database.....	86
Tabel 4.19	Atribut untuk tabel source database vendor .....	87
Tabel 4.20	Atribut untuk tabel source database tables .....	87
Tabel 4.21	Atribut untuk source database columns.....	85
Tabel 4.22	Pengukuran <i>precision</i> , <i>recall</i> dan akurasi.....	103
Tabel 4.23	Perbandingan ekstraksi sistem Rumah Sakit.....	111
Tabel 4.24	Perbandingan ekstraksi sistem Odoo ERP .....	115
Tabel 4.25	Perbandingan penelitian ekstraksi <i>event logs</i> .....	118

*Halaman ini sengaja dikosongkan*



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Kemudahan akses, kecepatan dan keakuratan saat ini menjadi sebuah dasar dalam semua sistem. Para pengembang teknologi informasi sekarang mulai mengembangkan sistem untuk mengefisiensi aktivitas pekerjaan untuk kebutuhan perusahaan atau instansi, beberapa contohnya yaitu: *e-commerce*, *hospital information system*, *academic information system*, *e-commerce*, *point of sales*, *customer relationship manajemen system* dan lain-lain. Dalam sebuah sistem proses bisnis merupakan bagian terpenting dalam membangun sistem yang baik. Secara luas proses bisnis saat ini didukung oleh sistem informasi. Salah satu bidang untuk mengelola dan mengoptimalkan bisnis proses adalah *Business Process Management* (BPM) (Jeston dan Nelis, 2014). Process mining adalah salah satu bagian penting dalam BPM yaitu terkait dengan Process Intelligence untuk mengevaluasi dan merancang ulang proses bisnis. Metode dalam BPM sekarang mulai beralih kearah *process mining* sebagai cara untuk membangun kembali atau meningkatkan keadaan yang ada menjadi lebih baik (Jans dkk., 2012). Penelitian tentang *process mining* masih banyak berkembang dan tergolong baru yang dikategorikan dalam bidang komputer cerdas dan data mining, permodelan proses dan analisis. Awal ide kemunculan dari *process mining* adalah untuk menemukan, memonitor dan meningkatkan proses nyata yang sudah terjadi dengan melakukan ekstrasi pengetahuan diambil dari *event logs* yang tersedia pada setiap sistem informasi saat ini (W. M. P. van der Aalst, 2011).

Pada bidang *process mining* beberapa teknik telah dikembangkan untuk memberikan aliran informasi dari pelaksanaan penggunaan sebuah sistem. Untuk menerapkan *mining analysis* dalam *process-aware information system* membutuhkan sebuah *event logs*. *Event logs* merupakan proses pencatatan histori berupa transaksi atau audit trail pada suatu *tool* sistem informasi (Mardhatillah dkk., 2012). Sebuah event logs berisi catatan log tentang aktivitas yang dieksekusi

untuk proses bisnis (Nakatumba dan van der Aalst, 2009). Event logs biasanya berisi data tentang case yang telah dilaksanakan dalam organisasi, waktu di mana task dieksekusi, orang atau sistem yang melakukan task ini, dan jenis lainnya dari data. Sebuah event logs memiliki beberapa atribut yaitu: Case ID (penanda sebuah proses dalam sistem), Task (proses yang dilakukan oleh pengguna sistem), Originator (pengguna yang melakukan task tertentu) dan *timestamp* (mengindikasikan waktu event untuk sebuah task).

Kualitas hasil dari *process mining* sangat bergantung pada masukannya (*event logs*). Maka *event logs* harus diperlakukan sebagai “*first-class citizen*” dalam sebuah sistem informasi untuk mendukung proses yang akan dianalisis. Atau dengan kata lain informasi yang terkandung dalam *event logs* harus mengandung informasi yang tepat dan berkualitas (W. M. P. van der Aalst, 2011). Oleh karena itu *process mining* sangat bergantung pada sumber data *event logs* sehingga kebutuhan akan ekstraksi *event logs* dari sebuah sistem menjadi penting untuk dilakukan. Beberapa sistem yang banyak digunakan instansi atau perusahaan tidak menyediakan secara langsung format *event logs* yang dibutuhkan oleh para analis bisnis. Contohnya pada sistem ERP (*Enterprises Resource Planning*) yang paling banyak digunakan di seluruh dunia tidak secara langsung menyediakan *event logs* yang dapat dipakai untuk kepentingan *process mining* (J. Buijs, 2010)

Salah satu alat yang digunakan pada analisa bidang *process mining* yaitu ProM Framework, pada versi terbaru nya alat tersebut dapat berjalan dengan masukan *event logs* dengan format XES (*eXtensible Event Stream*). XES sendiri diadopsi pada tahun 2010 oleh *IEEE Task Force on Process Mining* sebagai standart format untuk pencatatan *event*. Untuk mengkonversi data *event logs* dari penyimpanan data ke dalam format XES (Buijs, 2010) menciptakan sebuah alat tambahan yang bernama XESame. Aplikasi tersebut memetakan desain basis data sistem SAP berdasarkan proses bisnis yang ada, kemudian dibentuk menjadi data *event logs* dengan format XES yang sekarang telah diadopsi pada aplikasi ProMimport. Akan tetapi untuk menjalankan aplikasi tersebut membutuhkan pengetahuan lebih mengai struktur basis data yang digunakan dalam sebuah sistem dan juga pemahaman mengenai bahasa SQL (*Structured Query Language*).

Kemudian pada tahun berikutnya Piessens (2011) membuat sebuah alat ekstraksi *event logs* dengan antarmuka yang lebih sederhana untuk mengakomodir sistem SAP. Aplikasi tersebut berhasil menciptakan sebuah prototipe alat ekstraksi untuk proses pengadaan barang dan penjualan barang pada sistem SAP. Beberapa alat yang relevan dengan kebutuhan ekstraksi *event logs* yaitu EVS ModelBuilder (Ingvaldsen dan Gulla, 2007), ARIS Process Performance Manager (Scheer, 2006), serta SAP Solution Manager (SAP AG, 2004).

Sebagian besar informasi *event logs* sebuah perusahaan maupun instansi pemerintahan tersimpan dalam sebuah basis data relasional dalam sistem yang mereka jalankan (W. M. van der Aalst, 2015). Informasi tersebut tersimpan dalam kumpulan tabel dan kolom yang saling berelasi satu sama lain. Untuk mengambil informasi dalam basis data menggunakan bahasa SQL (*Structured Query Language*), tiap vendor DBMS memiliki struktur bahasa yang SQL berbeda. Sering kali seseorang yang ahli dalam bidang tertentu (Kesehatan, Sipil, Bisnis, Perbankan, dll) tidak memiliki pengetahuan lebih tentang bahasa komputer. Sehingga menyulitkan para ahli bidang tersebut untuk mengambil informasi yang mereka butuhkan dari sebuah basis data. Dengan kata lain untuk mengambil *event logs* yang dibutuhkan dalam process mining harus melibatkan Ahli IT. Melibatkan Ahli IT pastinya membutuhkan waktu dan biaya yang tidak sedikit, belum lagi jika Ahli IT tersebut tidak sepenuhnya memahami process mining dan *event log*. Oleh karena itu untuk meringankan permasalahan tersebut diperlukan sebuah media perantara antara ahli bidang dengan basis data agar memudahkan pengambilan informasi (dalam hal ini Event Log). Oleh karena itu pada penelitian ini mengusulkan dengan menggunakan bahasa alami yang sering digunakan sehari-hari. Bahasa alami sendiri merupakan bahasa yang secara umum digunakan oleh manusia dalam berkomunikasi satu dengan sama lain. Penggunaan bahasa alami dalam sistem komputer telah digunakan sejak tahun 1960, tujuan penggunaan bahasa alami tersebut agar manusia dapat berinteraksi dengan komputer. Bahasa yang diterima oleh komputer butuh untuk diproses dan dipahami terlebih dahulu agar maksud dari pengguna bisa dipahami dengan baik oleh komputer.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka munculah permasalahan baru yang akan dibahas penelitian ini sebagai berikut:

1. Seringkali analis mengalami kesulitan saat ingin mengambil data *event logs* yang tersimpan dalam basis data sebuah sistem. Diperlukan pemahaman pemahaman bahasa pemrograman dan bahasa SQL untuk melakukan itu dengan alat ekstraksi yang ada. Dengan memanfaatkan bahasa alami sebagai masukan akan mempermudah dalam memperoleh data *event logs* sesuai keinginan analis bisnis, namun sebuah susunan kata yang dibentuk menjadi kalimat dalam bahasa alami mengandung makna yang berbeda sehingga menimbulkan ambiguitas. Maka rumusan masalah yang pertama adalah: Bagaimana bahasa alami dapat menghasilkan sebuah perintah oleh analis bisnis sehingga dapat dirubah ke dalam bahasa SQL yang menghasilkan data *event logs* yang tersimpan dalam basis data?
2. Event logs merupakan sebuah catatan kejadian yang terjadi secara berurutan sesuai dengan case tertentu. Dengan demikian atribut waktu merupakan bagian utama dari sebuah event log. Rumusan masalah kedua adalah: Bagaimana cara bahasa alami yang digunakan dapat mengkonversi atribut waktu dalam *event logs* dan juga memberikan hasil event logs yang berurutan tanpa harus menggunakan bahasa SQL?
3. Kualitas hasil *process mining* sangat bergantung dari kualitas event log. Sebelum melanjutkan tahapan *process mining* Sayangnya beberapa alat ekstraksi yang ada jika terjadi kesalahan pada hasil event pengguna harus melakukan proses mulai dari awal, hal ini memakan waktu yang tidak sedikit. Oleh karena itu rumusan masalah ketiga adalah: Bagaimana ekstraksi dengan bahasa alami dapat menghasilkan *event logs* yang sesuai standar kemudian digunakan untuk kebutuhan *process mining* tanpa harus mengulangi dari awal saat terjadi kesalahan?



### 1.3 Batasan Penelitian

1. Jenis masukan yang digunakan pada penelitian terbatas pada basis data yang memiliki dimensi waktu dalam setiap transaksi di dalam sistem. Serta menggunakan DBMS PostgreSQL.
2. Proses ekstraksi menggunakan bahasa alami yang sudah terstruktur yang sudah ditentukan sebelumnya.
3. Jenis sistem dan penyimpanan data yang dijadikan ujicoba yaitu Sistem Informasi Manajemen Rumah Sakit pada proses bisnis permintaan barang, Odoo ERP pada proses bisnis penjualan.

### 1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Membuat sebuah prototipe alat/aplikasi ekstraksi *event logs* dengan kriteria sebagai berikut
  - Menggunakan antarmuka bahasa alami yang mengakomodir perintah ekstraksi *event logs* (mengandung atribut waktu) yang tersimpan dalam basis data serta dapat menghasilkan data informasi yang terurut sesuai catatan kejadian.
  - Memberikan hasil berupa data *event logs* yang akurat dan sesuai standart dalam proses yang lebih cepat yang selanjutnya dapat digunakan dalam kebutuhan process mining.

### 1.5 Lingkup Penelitian

Objek yang dijadikan penelitian ini adalah *sistem informasi* dengan melibatkan administrator sistem. Data yang digunakan pada penelitian ini adalah aktifitas proses bisnis dalam lingkup Sistem Informasi Manajemen yang memiliki proses bisnis dan aktivitas yang mengandung atribut waktu.

### 1.6 Kontribusi Penelitian

Kontribusi ilmiah yang akan dihasilkan dalam penelitian ini yaitu mengetahui pemanfaatan NLIDB dalam bidang *Business Process Management* serta hasil prototipe penelitian ini juga dapat dimanfaatkan oleh para analis bisnis ataupun manajemen perusahaan/instansi terkait yang ingin melakukan ekstraksi

*event logs* kemudian melakukan analisa *process mining* maupun aktivitas yang berjalan dalam sistem secara realtime.

## **1.7 Sistematika Penulisan**

Sistematika penulisan laporan penelitian yang dilakukan adalah sebagai berikut:

### **Bab I Pendahuluan**

Bab ini terdiri dari latar belakang dilakukannya penelitian, perumusan masalah, tujuan, batasan penelitian, kontribusi penelitian dan sistematika penulisan.

### **Bab II Kajian Pustaka**

Bab ini berisi kajian terhadap teori dan penelitian-penelitian yang sudah ada sebelumnya.

### **Bab III Metode Penelitian**

Dalam bab ini dijelaskan mengenai langkah-langkah penelitian beserta metode yang digunakan. Langkah-langkah penelitian dijelaskan dalam sebuah diagram alur yang sistematis dan akan dijelaskan tahap demi tahap.

### **Bab IV Pembahasan**

Dalam bab ini akan dilakukan perancangan aplikasi kemudian dilakukan uji coba terhadap alat ekstraksi yang telah dirancang sebelumnya. Uji coba ini dilakukan berdasarkan skenario uji coba yang telah dirancang sebelumnya. Selain itu dalam bab ini juga dijelaskan mengenai analisis hasil uji coba.

### **Bab V Kesimpulan dan Saran**

Bab ini berisi kesimpulan dari penelitian ini dan juga saran bagi penelitian mendatang yang berasal dari kekurangan ataupun temuan dari penelitian ini.

### **Daftar Pustaka**

Berisi daftar referensi yang digunakan dalam penelitian ini, baik jurnal, buku maupun artikel.

*Halaman ini sengaja dikosongkan*



## **BAB 2**

### **KAJIAN PUSTAKA DAN LANDASAN TEORI**

Pada bab ini akan dibahas mengenai kajian pustaka yang diambil dari penelitian-penelitian sebelumnya yang relevan. Kajian pustaka ini selanjutnya akan digunakan sebagai landasan dalam melakukan penelitian ini

#### **2.1 Penelitian Terdahulu**

Studi literatur dalam penelitian ini digunakan untuk mendapatkan gambaran yang menyeluruh tentang apa yang sudah dikerjakan peneliti lain dan bagaimana peneliti mengerjakannya, kemudian mengidentifikasi celah penelitian sebagai dasar penelitian yang akan dilakukan. Penggalan informasi dari literatur dilakukan sebagai upaya memperjelas permasalahan yang memiliki kaitan dengan penelitian ini, sekaligus untuk membedakan antara penelitian ini dengan penelitian sebelumnya. Literatur yang digunakan sebagai acuan dalam penelitian ini merupakan hasil penelitian dari kalangan akademis yang disajikan dalam bentuk paper dalam jurnal, paper hasil seminar dan tesis.

Ekstraksi *event logs* dianggap sebagai langkah penting dalam *process mining*. Struktur dan isi dari sebuah *event logs* menggambarkan hasil yang dapat diperoleh dalam tahap *process mining*. Langkah pertama yang penting dalam prosedur ekstraksi *event logs* adalah untuk membuat beberapa keputusan tentang tujuan dalam lingkup analisa dalam *process mining* itu sendiri. Hal ini membantu dalam memetakan proses bisnis untuk dianalisis dan menghindari masalah yang terjadi kemudian. Beberapa penelitian terkait ekstraksi *event log* telah dilakukan diantaranya (de Murillas dkk., 2015) dengan memanfaatkan *redo log* yang disediakan oleh vendor DBMS. Kemudian dengan menggunakan pendekatan ontologi (Calvanese dkk., 2015) yang memberikan paradigma baru untuk proses ekstraksi *event logs*. Selain itu agar mempermudah orang awam untuk mengenal *process mining* dan tata cara pengambilan ekstraksi *event logs* dalam sebuah sistem (W. M. van der Aalst, 2015) membuat sebuah guideline yang rinci serta memodelkan sebuah data kejadian yang secara umum tersimpan dalam sebuah basis data dari sebuah sistem. Penelitian terkait ekstraksi *event logs* masih sedikit

dilakukan oleh para peneliti, hal ini disebabkan proses ekstraksi *event logs* membutuhkan waktu dan tenaga besar. Selain itu alat yang telah dikembangkan untuk proses ekstraksi event log akan tetapi masih banyak keterbatasan serta tidak ada lagi penelitian yang menutupi keterbatasan dari penelitian yang ada.

Untuk mendukung tugas rutin mengubah sumber data dari berbagai sistem ke format MXML, kerangka kerja ProM Import dibuat (Günther dan van der Aalst, 2006). Tujuan utama dari kerangka kerja ini adalah kemampuan penggunaan yang berkelanjutan dengan memanfaatkan plug-in untuk format sumber tertentu sambil tetap menyediakan fungsionalitas umum dari kerangka kerja. Aspek penting lainnya adalah efisiensi penulisan file MXML sementara *event logs* dapat diambil dari sumberdata tetapi tidak sesuai urutan penulisannya. Meskipun ProM Import telah terbukti sangat berguna dalam mengubah log dari sistem sebenarnya ke format MXML, namun juga memiliki beberapa keterbatasan. Dengan memungkinkan keberlanjutan penggunaan penggunaan plug-in yang ditulis dengan bahasa Java, konversi baru hanya dapat dibuat oleh orang-orang yang mengetahui struktur Kerangka kerja ProM Import dan memiliki pengetahuan pemrograman pada bahasa Java.

The Enterprise Visualization Suite (EVS) memungkinkan penerapan kombinasi *process mining*, pengumpulan data dan fungsi statistik pada siklus kejadian (Ingvaldsen, 2011). Siklus kejadian adalah interpretasi jejak yang lebih general. Peristiwa dalam rangkaian kejadian tidak harus berhubungan dengan *process instance* tunggal. Alasan utama untuk tidak menghubungkan kejadian dengan *process instance* tunggal karena kompleksitas sistem informasi seperti SAP. Dalam sistem ini tidak selalu ada pemetaan yang jelas antara kejadian dan proses kejadian. Meskipun Model EVS saat ini secara khusus ditujukan untuk mengekstraksi rangkaian kejadian dari database SAP, prinsip yang sama dapat diterapkan ke sumber data apa pun yang didasarkan pada penyimpanan basis data lainnya. Salah satu faktor menyulitkan untuk menggunakan EVS ModelBuilder untuk mengekstrak *event logs* MXML atau XES dari sumber data adalah tidak adanya hubungan antara kejadian dan contoh proses tunggal. Selanjutnya, setiap kejadian perlu didefinisikan secara eksplisit.

ARIS Process Performance Manager (PPM) adalah produk yang dikeluarkan oleh IDS Scheer. Produk ini merupakan bagian dari platform ARIS dan berkontribusi pada solusi pengelolaan SAP berbasis proses (Scheer, 2006). Keuntungan dari toolset ARIS adalah yang memiliki hubungan yang erat dengan SAP. PPM dapat memvisualisasikan bagaimana proses dijalankan dengan menggunakan data langsung, dan dapat merekonstruksi pelaksanaan setiap transaksi bisnis dari awal sampai akhir. PPM tersebut merupakan alat yang baik untuk analisa proses bisnis dalam SAP akan tetapi secara spesifik tidak dapat mendukung format XES untuk process mining.

Dalam prototipe aplikasinya, XES Mapper, memungkinkan pengguna dapat menentukan pernyataan SQL setiap tindakan, yaitu atribut dan property yang menjadi bagian suatu kejadian tertentu. Dalam sistem SAP, kejadian yang menyertai aktivitas tertentu tersimpan dalam basis data dan hanya bisa diambil dengan menggunakan SQL. Prototipe XES Mapper telah dikembangkan lebih lanjut oleh Buijs dan disertakan sebagai XESame dalam toolkit ProM 6 ( i J. C. A. M. Buijs, 2010) XESame memungkinkan pakar domain mengekstrak log aktivitas dari sistem informasi yang ada tanpa harus memprogram. Namun untuk mengekstrak informasi *event logs* dari sistem selain SAP membutuhkan pengetahuan dalam basis data untuk sistem tersebut serta membutuhkan waktu lagi untuk mengkonstruksi kembali atribut dan properti dalam sistem tersebut.

Untuk menutupi kekurangan pada ( i J. C. A. M. Buijs, 2010) maka (Piessens, 2011) melakukan penelitian dengan judul “*Event Log Extraction from SAP ECC 6.0*” yang menghasilkan prototipe alat ekstraksi baru yang fokus pada proses pengadaan barang dan penjualan barang sistem SAP. Alat tersebut memiliki antarmuka yang mudah digunakan serta tidak perlu memiliki pengetahuan lebih dalam struktur basis data SAP. Selanjutnya hasil alat tersebut juga dapat digunakan pada platform process mining bernama Futura Process Intelligence.

Dari beberapa literatur diatas keterbatasan ekstraksi *event logs* diantaranya: hanya dapat dilakukan pada sistem SAP, dapat dilakukan untuk proses tertentu, pengambilan *event logs* tidak bisa dilakukan secara realtime, membutuhkan pengetahuan lebih soal struktur penyimpanan data dalam sebuah sistem dan yang

terakhir membutuhkan pengetahuan tentang menjalankan bahasa SQL dalam DBMS.

Salah satu cara mengambil informasi dari basis data tanpa menggunakan bahasa SQL adalah dengan NLIDB (Natural Language Interface to A Database), yaitu dengan memanfaatkan bahasa alami yang sering digunakan manusia untuk komunikasi sebagai masukan dengan keluaran hasil informasi yang dibutuhkan sesuai keinginan. Didalam bahasa Indonesia NLIDB sudah digunakan untuk mengkonversi bahasa natural ke dalam query database (Andayani, 2002) dengan melakukan pengambilan kembali informasi yang berasal dari database akademik. Bahasa Indonesia sudah mempunyai grammar dan aturan produksi. Elemen pemroses bahasa terdiri dari penganalisis leksikal, parser dan pembangkit kode atau perlakuan. Maier dan Warren dalam (Hartati dan Zuliarso, 2011) menyatakan komponen pemrosesan bahasa alami terdiri dari scanner, parser, penterjemah (translator), optimasi query dan pengevaluator query.

Beberapa penelitian terbaru terkait NLIDB dalam bahasa indonesia yaitu (Wibisono, 2013) menggunakan basis data XML sebagai sumber data lalu menggunakan XQuery untuk menterjemahkan bahasa natural ke dalam sumber data, (Raharjo dan Hartati, 2014) untuk menerjemahkan bahasa natural ke dalam query query terhadap basis data terjemahan Al-Quran, (Reinaldha dan Widagdo, 2014) mampu menangani sebuah pertanyaan dan konversi satuan unit. Dari penelitian yang ada penerapan NLIDB dapat memberikan hasil yang baik dan akurat dalam menangani menerjemahkan bahasa alami ke dalam query database.

Dalam sebuah sistem informasi terdapat data transaksi yang seringkali berkaitan dengan waktu. Contohnya adalah data pembayaran, data pembelian, data pengembalian barang, data peminjaman dan data lain yang berkaitan sebuah transaksi konsumen. Pada penelitian yang dilakukan (Reinaldha dan Widagdo, 2014) pengujian pertanyaan yang berkaitan dengan waktu akan muncul sebuah error yang berarti penelitian tersebut masih belum bisa menangani jika terdapat atribut waktu dalam masukannya. Oleh karena celah ini dapat menjadi masukan untuk penelitian baru dibidang NLIDB bahasa indonesia.



## **2.2 Proses Bisnis**

Proses bisnis terdiri dari kumpulan tugas atau kegiatan terstruktur yang saling terkait dalam sebuah organisasi untuk memecahkan masalah tertentu. Sebuah proses bisnis merupakan kombinasi dari serangkaian kegiatan dalam suatu perusahaan dengan struktur yang menggambarkan urutan logis dan ketergantungan yang bertujuan untuk memperoleh hasil yang diinginkan (Aguilar-Saven, 2004). Suatu proses bisnis tidak hanya menghubungkan semua kegiatan dalam suatu organisasi, tetapi dapat juga saling berinteraksi dengan proses bisnis yang dilakukan oleh organisasi lain.

Proses bisnis dapat didefinisikan sebagai serangkaian kegiatan yang dieksekusi sesuai dengan aturan-aturan tertentu untuk mencapai tujuan bisnis yang telah ditetapkan (Betz dkk., 2011). Pengertian proses bisnis menurut (Laguna dan Marklund, 2013) adalah jaringan aktivitas yang terhubung dengan batas-batas yang jelas, yang memanfaatkan sumber daya untuk mengubah input menjadi output untuk tujuan memuaskan kebutuhan pelanggan.

Sebuah proses bisnis terdiri dari satu set tugas yang saling terkait yang bergabung bersama dengan elemen control flow (Guo dkk., 2008). Ada enam karakteristik yang dimiliki oleh sebuah proses bisnis, antara lain (Greg Alexander dkk., 2008):

1. Definability: Sebuah proses bisnis harus memiliki batasan, masukan (input), dan keluaran (output) yang jelas. Input merupakan permintaan dari pelanggan dan output merupakan hasil dari proses. Secara umum, antara input awal dan output akhir, interaksi tambahan antara pemohon layanan dan organisasi penyedia jasa dapat diperlukan.
2. Order: Sebuah proses bisnis harus terdiri dari aktivitas yang diurutkan berdasarkan waktu dan ruang. Aktivitas ini dilaksanakan oleh partisipan dari suatu proses yang terlibat pada fungsi, organisasi, atau perusahaan yang berbeda.
3. Customer: Sebuah proses bisnis harus mempunyai sasaran penerima hasil akhir proses atau pelanggan.

4. Value Adding: Transformasi yang terjadi dalam proses harus memberikan nilai tambah pada penerima.
5. Embeddedness: Sebuah proses tidak dapat berdiri sendiri, melainkan harus terkait dalam suatu struktur organisasi. Dalam ekonomi global, sebagian besar proses menjangkau beberapa fungsi atau bahkan perusahaan dengan menugaskan eksekutor yang sesuai. Eksekutor bisa berupa manusia atau mesin seperti perangkat lunak ERP.
6. Cross-functionality: Sebuah proses umumnya mencakup beberapa fungsi. Namun, hal ini tidak harus berlaku demikian. Contoh kerjasama Cross functionality meliputi negosiasi kontrak (dengan legal), pengumpulan intelijen kompetitif (dengan pemasaran), upselling pelanggan yang sudah ada (dengan dukungan pelanggan), dan meningkatkan interaksi klien dan otomatisasi (teknologi informasi).

Suatu proses bisnis dapat dibagi menjadi beberapa subproses yang masing-masing memiliki atribut sendiri dan juga berkontribusi untuk mencapai tujuan dari superprosesnya (Kennedy dan Sharma, 2009). Menurut (Poonia, 2010) ada tiga jenis proses bisnis, di antaranya:

1. Management Processes, yaitu proses yang mengendalikan operasional dari sebuah sistem, termasuk Corporate Governance dan Strategic Management.
2. Operational Processes, yaitu proses yang merupakan bisnis inti dan menciptakan aliran nilai utama, termasuk Purchasing, Manufacturing, Marketing dan Sales.
3. Supporting Processes, yaitu proses yang mendukung proses inti. Contohnya termasuk Accounting, Recruitment, dan Technical Support.

Proses bisnis merupakan salah satu sumber penting untuk penyelenggaraan daya saing bisnis (Marquez dkk., 2014). Pelaksanaan dan penerapan proses bisnis dapat memberikan keunggulan kompetitif bagi perusahaan. Menurut (Jung dkk., 2008) proses bisnis dengan kumpulan tugas terstandarisasi dan terstruktur dapat mendorong penciptaan nilai dari suatu perusahaan. Perusahaan dapat

meningkatkan efektivitas produksi dan operasionalnya dengan proses bisnis yang telah terstandarisasi.

### **2.3 Process Mining**

Process mining dapat dilihat sebagai link yang mengisi ruang antara data mining dan *Business Process Management* (BPM). Teknologi Process mining menggabungkan kekuatan dari data mining dan aplikasi pemodelan proses. Proses mining merupakan suatu teknologi yang relatif masih baru dalam kaitannya dengan BPM. BPM sendiri bertujuan untuk mendapatkan model dengan cara mengamati perilaku proses bisnis di suatu organisasi. Pada proses mining, pengamatan dilakukan terhadap proses bisnis yang telah terkomputerisasi.

Process mining adalah sebuah teknik yang menggunakan catatan kejadian (event log) untuk menganalisis pelaksanaan proses bisnis atau workflow (Van Der Aalst dan Van Hee, 2004). Process mining bertujuan untuk menemukan model proses dari event log yang dicatat oleh sistem informasi (Wang dkk., 2012). Process mining dimulai dengan informasi tentang proses yang dieksekusi dan dikumpulkan oleh sistem informasi, yaitu sistem transaksional atau Process Aware Information System (PAIS), seperti Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Business-to-business (B2B) dan sistem Workflow Management (WFM). Sistem informasi ini menawarkan informasi tentang urutan dimana event dari sebuah case dieksekusi.

Process mining dapat digambarkan sebagai proses dengan tiga tahap: pre-processing, processing dan post-processing (Medeiros, 2004). Pada tahap pre-processing, didasarkan pada asumsi bahwa log yang menjadi input berisi informasi yang cukup, sehingga ordering relation dari task dapat disimpulkan. Tahap processing sesuai dengan eksekusi algoritma pertambangan, dengan log dan ordering relation sebagai input. Selama post-processing, model yang ditemukan (misalnya, Petri net) dapat disempurnakan dan representasi grafis dapat dibangun.

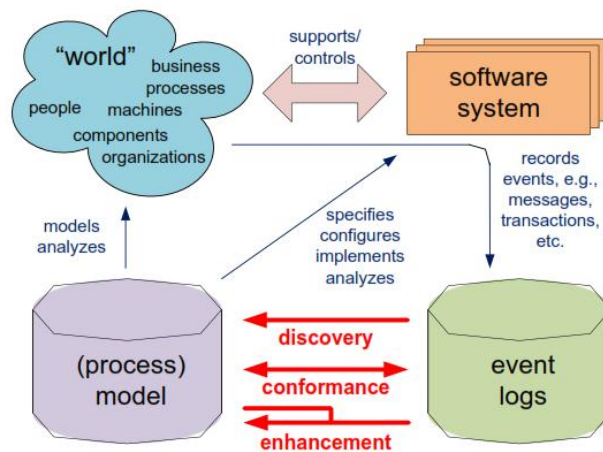
Process mining bertujuan menggali informasi dari event log untuk mendapatkan proses bisnis yang sedang dijalankan (Van Der Aalst dan Van Hee, 2004). Karakteristik Process mining menurut (van der Aalst dkk., 2012), adalah:

1. Process mining tidak terbatas pada penemuan control-flow. Penemuan control-flow sering dianggap sebagai bagian yang paling menarik dari Process mining. Di sisi lain, penemuan (discovery) hanyalah salah satu dari tiga bentuk dasar Process mining (discovery, conformance, dan enhancement). Ruang lingkup Process mining tidak hanya terbatas pada control-flow, tetapi perspektif organisasi, kasus dan waktu juga berperan penting.
2. Process mining bukan hanya jenis tertentu dari data mining. Process mining dapat dilihat sebagai "missing link" antara data mining dan *model driven* BPM tradisional. Kebanyakan teknik data mining tidak *process-centric* sama sekali. Proses model berpotensi menunjukkan konkurensi yang tak tertandingi untuk struktur data mining sederhana seperti decision trees dan association rules. Oleh karena itu, jenis representasi dan algoritma yang baru benar-benar diperlukan.
3. Process mining tidak terbatas pada analisis offline. Teknik Process mining mengekstrak pengetahuan dari kejadian historis. Meskipun data "post mortem" yang digunakan, hasilnya dapat diterapkan untuk menjalankan case. Misalnya, waktu penyelesaian pesanan pelanggan yang sebagian ditangani dapat diprediksi dengan menggunakan model proses yang ditemukan.

Objektivitas merupakan keuntungan utama dari Process mining karena penggunaan data nyata dalam analisis. Manfaat lain dari Process mining adalah menemukan potensial exception dan juga membantu untuk mengidentifikasi penyebab exception tersebut (Saylam dan Sahingoz, 2013). Teknik Process mining dengan menggunakan event log secara otomatis dapat menemukan model proses, memeriksa kesesuaian (Conformance checking), mengidentifikasi hambatan (bottlenecks) dan penyimpangan (deviasi), mengusulkan perbaikan, dan memprediksi waktu pengolahan (Bose dkk., 2013).

Process mining terdiri dari kumpulan teknik untuk menganalisis informasi yang tersimpan dalam event log, di mana analisis berfokus pada penemuan

(discovery), pemantauan (monitoring) dan perbaikan (enhancement) proses. Menurut (Khodabandelou dkk., 2013) mengklasifikasikan teknik Process mining menjadi tiga kategori, yaitu Discovery, Conformance, dan Enhancement seperti dalam Gambar 2.1



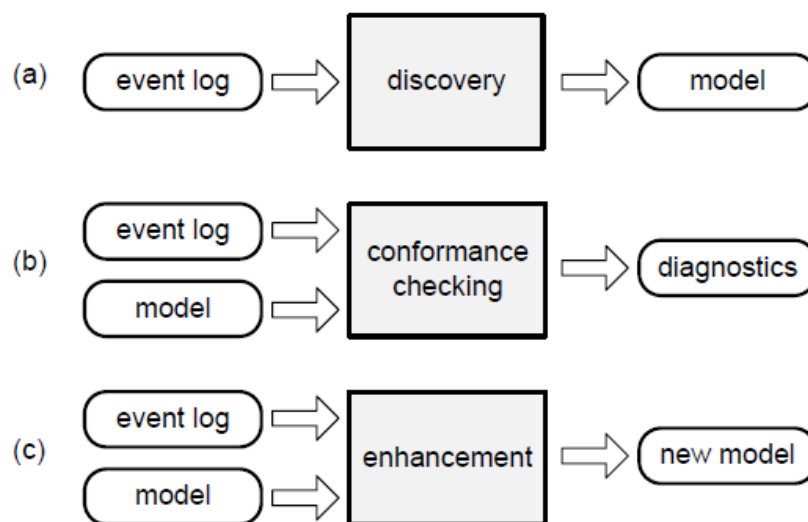
**Gambar 2.1** Tiga Jenis Utama dari Process Mining : (a) Discovery, (b) Conformance, (c) Enhancement (Aalst, dkk., 2012)

Discovery merupakan teknik yang bertujuan untuk menemukan model proses dengan menganalisis event log. Model BPMN dan Petri net dapat diberikan sebagai contoh teknik untuk penemuan proses. Proses ini tidak hanya menemukan control-flow dari sistem, tetapi juga model organisasi. Misalnya, event log yang ditambah dengan algoritma alpha menghasilkan model Petri net yang menyajikan perilaku para aktor yang dicatat dalam event log.

Conformance merupakan teknik dimana model proses yang ada (apriori model) dibandingkan dengan model yang dihasilkan oleh event log dari proses yang sama. Teknik ini dapat digunakan untuk memeriksa apakah realitas, seperti yang tercatat dalam log, sesuai dengan model dan sebaliknya. Selain itu untuk mengidentifikasi perbedaan antara apa yang diharapkan terjadi dan apa yang sebenarnya terjadi. Teknik ini bertujuan mendeteksi inkonsistensi dan/atau penyimpangan (tentang siapa, apa, kapan dan di mana) antara model proses dari file log. Berbagai jenis model yang berbeda dapat digunakan untuk memeriksa konsistensi model proses aktual terhadap model proses yang telah ditetapkan

(conformance checking), diantaranya model prosedural, model organisasi, model deklaratif proses, aturan bisnis atau kebijakan, hukum, dan lain-lain.

Enhancement merupakan teknik yang menggunakan informasi yang terekam dalam event log untuk meningkatkan dan memperkaya model proses yang sebenarnya menggunakan metode perbaikan (repair) dan perluasan (extension). Repair memiliki efek cermin, yaitu mencoba untuk membentuk kembali model untuk lebih menggambarkan realitas. Extension memungkinkan perluasan model proses dengan aspek baru berdasarkan korelasi silangnya dengan log. Misalnya, dengan menggunakan timestamp dalam event log yang dapat memperluas model untuk menunjukkan hambatan (bottlenecks), tingkat layanan (service levels), throughput times, dan frekuensi.



**Gambar 2.2** Tiga Tipe Dasar Process mining yang Dijelaskan dalam Hal Input dan Output: (A) Discovery, (B) Conformance Checking, dan (C) Enhancement  
(W. M. P. van der Aalst dkk., 2012)

## 2.4 Event Log

Sistem informasi biasanya mendukung kemampuan *logging* yang mendaftarkan apa yang telah dieksekusi dalam organisasi. Sistem informasi transaksional, seperti Workflow Management System (WMS), Customer Relationship Management (CRM), Supply Chain Management (SCM), Enterprise

Resource Planning (ERP) atau sistem B2B dapat digunakan oleh perusahaan untuk mendukung pelaksanaan proses bisnis. Setiap sistem informasi transaksional memiliki event log sebagai bukti dari transaksi dan untuk memonitor proses yang sedang berjalan.

Event log merupakan proses pencatatan histori berupa transaksi atau audit trail pada suatu tool sistem informasi (Mardhatillah dkk., 2012). Sebuah event log berisi catatan log tentang aktivitas yang dieksekusi untuk proses bisnis (Nakatumba dan van der Aalst, 2009). Event log biasanya berisi data tentang case yang telah dilaksanakan dalam organisasi, waktu di mana task dieksekusi, orang atau sistem yang melakukan task ini, dan jenis lainnya dari data. Menurut (Wang dkk., 2011) event log berisi informasi tentang event yang mengacu pada suatu case dan task tertentu.

Pada event log, setiap event sesuai dengan task yang dieksekusi dalam proses, beberapa event dihubungkan bersama dalam process instance atau case, dan setiap case membentuk urutan event yang diperintahkan oleh timestamp mereka. Case (yang juga bernama process instances) merupakan sesuatu hal yang sedang dikerjakan. Contoh dari case adalah order dari pelanggan, klaim asuransi, dan penanganan keluhan pelanggan. Satu case bisa terdiri dari banyak task. Task (biasa dikenal dengan istilah Activity, operation, action, atau work-item) adalah beberapa operasi dari case. Urutan task yang tercatat selama eksekusi process instance disebut workflow trace (Walicki dan Ferreira, 2011).

Event log seperti yang ditunjukkan pada Tabel 2.2 digunakan sebagai starting point untuk Process mining. Event log tersebut berisi informasi tentang empat process instance (case) dari sebuah proses yang menangani denda (fine). Berdasarkan baris yang berwarna abu-abu, dapat dilihat satu process instance dengan Case ID 1 yang dimulai dengan status File Fine pada tanggal 20 Juli 2004, berpindah ke Send Bill dan Process Payment, dan berakhir dengan statusnya Close Case pada tanggal 24 Juli 2004. Berdasarkan Tabel 2.2 dapat diketahui atribut-atribut yang dimiliki oleh sebuah event log, diantaranya:

1. Case ID: Sebuah identifier case yang diperlukan untuk membedakan eksekusi yang berbeda dari proses yang sama. Misalnya, di sebuah

pusat panggilan, case ID mungkin dapat berupa nomor permintaan layanan dan di rumah sakit, case ID mungkin dapat berupa ID pasien.

2. Task: Pada task, harus ada nama untuk langkah-langkah proses yang berbeda atau perubahan status yang dilakukan dalam proses. Jika hanya memiliki satu entri (satu baris) untuk setiap process instance, maka data tersebut dapat dikatakan tidak cukup rinci.
3. Originator: Pengguna yang melakukan task tertentu atau orang yang melaksanakan atau memulai event.
4. Timestamp: Satu timestamp diperlukan untuk membawa event dalam urutan yang benar. Timestamp digunakan untuk mengindikasikan waktu event untuk sebuah task, mengidentifikasi penundaan antara task dan mengidentifikasi hambatan dalam proses.

**Tabel 2.1** Contoh Event Log

Case ID	Task Name	Event Type	Originator	Timestamp	Extra Data
1	File Fine	Completed	Anne	20-07-2004 14:00:00	...
2	File Fine	Completed	Anne	20-07-2004 15:00:00	...
1	Send Bill	Completed	system	20-07-2004 15:05:00	...
2	Send Bill	Completed	system	20-07-2004 15:07:00	...
3	File Fine	Completed	Anne	21-07-2004 10:00:00	...
3	Send Bill	Completed	system	21-07-2004 14:00:00	...
4	File Fine	Completed	Anne	22-07-2004 11:00:00	...
4	Send Bill	Completed	system	22-07-2004 11:10:00	...
1	Process Payment	Completed	system	24-07-2004 15:05:00	...
1	Close Case	Completed	system	24-07-2004 15:06:00	...
2	Send Reminder	Completed	Mary	20-08-2004 10:00:00	...
3	Send Reminder	Completed	John	21-08-2004 10:00:00	...
2	Process Payment	Completed	system	22-08-2004 09:05:00	...
2	Close case	Completed	system	22-08-2004 09:06:00	...
4	Send Reminder	Completed	John	22-08-2004 15:10:00	...
4	Send Reminder	Completed	Mary	22-08-2004 17:10:00	...
4	Process Payment	Completed	system	29-08-2004 14:01:00	...
4	Close Case	Completed	system	29-08-2004 17:30:00	...
3	Send Reminder	Completed	John	21-09-2004 10:00:00	...
3	Send Reminder	Completed	John	21-10-2004 10:00:00	...
3	Process Payment	Completed	system	25-10-2004 14:00:00	...
3	Close Case	Completed	system	25-10-2004 14:01:00	...

#### 2.4.1 Event log pada sistem informasi

Seperti pada pembahasan sebelumnya sebuah *event log* mengandung catatan sebuah aktivitas. Secara umum catatan aktivitas tersebut berkaitan dengan contoh proses atau kejadian tertentu dalam sebuah sistem. Oleh karena itu tiap catatan berisi peristiwa yang muncul dalam sebuah contoh proses.

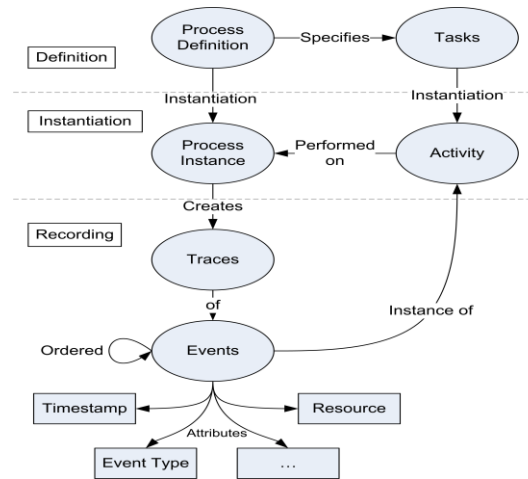


Sebuah contoh proses seringkali dijadikan sebagai *business objects*. *Business Objects* adalah obyek atau item yang digunakan oleh bisnis. Contohnya pasien, mesin robot, pembelian, invoice dan klaim asuransi. Seringkali catatan dianggap sebagai obyek yang kurang nyata contohnya perawatan dalam rumah sakit, penggunaan mesin saat menyala maupun mati, sesi pengguna saat mengakses website, pembicaraan dalam email dan lain-lain. Secara umum *business object* disimpan dalam sebuah table yang disebut data master. Master data ini merekam informasi tentang *business object* dan sebagai tambahan atribut pada sebuah catatan. Sebagai gambaran terdapat sebuah catatan pembelian barang pada sebuah perusahaan maka dalam sebuah sistem, informasi barang yang dibeli tersebut tidak disimpan pada pencatatan aktivitas pembelian akan tetapi disimpan didalam master data barang.

Hal ini penting untuk diingat bahwa sebuah event log hanya terdiri dari satu contoh proses yang terkait dengan sebuah kejadian. Misalnya *case* pada *event* tertentu di *event log* tidak bisa hanya berhubungan dengan pasien sementara *event* lainnya berkaitan dengan mesin seperti kegiatan pemeliharaan mesin. Oleh karena itu sebuah catatan harus berisi *event* yang berkaitan pada satu *business object*.

Secara umum *event log* mencatat peristiwa yang terjadi dalam proses tertentu untuk kasus tertentu. Pada gambar 2.3 mendeskripsikan struktur event log yang umumnya ada pada sistem informasi. Pada *process definition* menentukan aktivitas yang akan dijalankan dalam sebuah struktur. Ketika sebuah *case* dimulai sebuah proses baru akan dihasilkan yang disebut *process instance*. Pada *process instance* tersebut melacak status terkini dari *case* dalam sebuah proses. Tiap kejadian adalah turunan dari aktifitas tertentu yang sudah didefinisikan dalam *process definition*. Selain itu, kejadian diurutkan untuk menunjukkan di mana urutan aktifitas terjadi. Dalam kebanyakan kasus aturan ini ditentukan oleh tanggal dan waktu atau *timestamp* sebagai atribut dari *event*. Terkadang informasi mulai dan selesai dicatat dari aktivitas tunggal. Hal ini tercatat dalam *event type* pada atribut *event*. Atribut lain yang umum adalah *resource* yang melakukan eksekusi *event* yang dapat menjadi pengguna dari sistem, sistem itu sendiri atau sistem eksternal. Banyak atribut lainnya dapat disimpan dalam log event yang

berhubungan dengan *event* misalnya atribut data yang ditambahkan atau diubah dalam sebuah aktivitas.



**Gambar 2.3** Struktur Umum *Event Log*

## 2.5 Natural Language Processing (NLP)

Dalam filsafat bahasa, bahasa alami atau bahasa natural adalah suatu bahasa yang diucapkan, ditulis, atau diisyaratkan (secara visual atau isyarat lain) oleh manusia untuk komunikasi umum. Bahasa jenis ini dibedakan dengan bahasa formal seperti bahasa pemrograman komputer atau "bahasa" yang digunakan dalam kajian logika formal, terutama logika matematika - serta bahasa buatan. Natural language adalah bahasa yang dapat dipahami dan dimengerti oleh individu pada lingkungan tertentu. Bahasa merupakan fenomena yang rumit yang melibatkan proses pengenalan bunyi, sintaksis kalimat serta inferensi semantik tingkat tinggi.

*Natural Language Processing* atau Pemrosesan Bahasa Alami merupakan salah satu tujuan jangka panjang dari *Artificial Intelligence* (kecerdasan buatan) yaitu pembuatan program yang memiliki kemampuan untuk memahami bahasa manusia. Pada prinsipnya bahasa alami adalah suatu bentuk representasi dari suatu pesan yang ingin dikomunikasikan antar manusia. Bentuk utama representasinya adalah berupa suara/ucapan, tetapi sering pula dinyatakan dalam bentuk tulisan. Inti dari pemrosesan bahasa alami adalah penguraian kalimat atau

sering disebut dengan parser. Parser berfungsi untuk membaca kalimat, kata demi kata dan menentukan jenis kata apa saja yang boleh mengikuti kata tersebut.

Pengolahan bahasa alami (Natural Language Processing - NLP) didefinisikan sebagai kemampuan suatu komputer untuk memproses bahasa, baik lisan maupun tulisan yang digunakan oleh manusia dalam percakapan sehari-hari. Untuk proses komputasi bahasa harus direpresentasikan sebagai suatu rangkaian simbol yang memenuhi aturan tertentu. Dalam pemrosesan bahasa alami terdapat beberapa kesulitan diantaranya sering terjadi ambiguity atau makna ganda dan jumlah kosa kata dalam bahasa alami besar dan berkembang dari waktu ke waktu.

### **2.5.1 Bidang pengetahuan dalam NLP**

Secara singkat pengolahan bahasa alami (natural language processing) mengenal beberapa tingkatan pengolahan yaitu:

1. Fonetik dan fonologi, berhubungan dengan suara yang menghasilkan kata yang dapat dikenali. Bidang ini menjadi penting dalam proses aplikasi yang memakai metode speech based system.
2. Morfologi, yaitu pengetahuan tentang kata dan bentuknya dimanfaatkan untuk membedakan satu kata dengan lainnya. Pada tingkat ini juga dapat dipisahkan antara kata dan elemen lain seperti tanda baca.
3. Sintaktis, yaitu pemahaman tentang urutan kata dalam pembentukan kalimat dan hubungan antar kata tersebut dalam proses perubahan bentuk dari kalimat menjadi bentuk yang sistematis. Meliputi proses pengaturan tata letak suatu kata dalam kalimat akan membentuk kalimat yang dapat dikenali. Selain itu dapat pula dikenali bagian-bagian kalimat dalam suatu kalimat yang lebih besar.
4. Semantik, yaitu pemetaan bentuk struktur sintaksi dengan memanfaatkan tiap kata ke dalam bentuk yang lebih mendasar dan tidak tergantung struktur kalimat. Semantik mempelajari arti suatu kata dan bagaimana arti dari kata tersebut membentuk suatu arti dari

kalimat yang utuh. Dalam tingkatan ini belum tercakup konteks dari kalimat tersebut.

5. Pragmatik, pengetahuan pada tingkatan ini berkaitan dengan masing-masing konteks yang berbeda tergantung pada situasi dan tujuan pembuatan sistem
6. Discourse knowledge, melakukan pengenalan apakah suatu kalimat yang sudah dibaca dan dikenali sebelumnya akan mempengaruhi arti dari kalimat selanjutnya. Informasi ini penting diketahui untuk melakukan pengolahan arti terhadap kata ganti orang dan untuk mengartikan aspek sementara dari informasi.
7. World knowledge, mencakup arti sebuah kata secara umum dan apakah ada arti khusus bagi suatu kata dalam suatu percakapan dengan konteks tertentu.

Definisi ini tidaklah bersifat baku, dan untuk setiap bentuk bahasa alami yang ada biasanya ada pendefinisian lagi yang lebih spesifik sesuai dengan karakter bahasa tersebut. Pada beberapa masalah mungkin hanya mengambil beberapa dari pendekatan tersebut bahkan mungkin ada yang melakukan tambahan proses sesuai dengan karakter dari bahasa yang digunakan dan sistem yang dibentuk.

## **2.6 Natural Language Interface to a Database (NLIDB)**

NLIDB merupakan untuk menerima permintaan dalam bahasa Inggris atau bahasa alam lainnya dan mencoba untuk 'memahami' mereka atau kita dapat mengatakan bahwa Natural Language Interface to Database (NLIDB) adalah sistem yang menerjemahkan kalimat bahasa alami ke dalam query database. Meskipun penelitian awal telah dimulai sejak akhir tahun enam puluhan, NLIDB tetap sebagai masalah penelitian terbuka. Sebuah sistem NLIDB lengkap akan menguntungkan kita dalam banyak cara. Secara tradisional, orang bekerja dengan bentuk harapan mereka bergantung pada atribut dalam sebuah formulir. Maka kehadiran NLIDB membuat seluruh pendekatan menjadi lebih fleksibel, karena dapat memaksimalkan penggunaan database yang biasanya tidak semua orang bisa memperoleh informasi didalamnya.

### 2.6.1 Sub Komponen NLIDB

Ilmuwan komputer telah membagi masalah akses NLIDB menjadi dua sub-komponen

- **Komponen Linguistik**

Hal ini bertanggung jawab untuk menerjemahkan masukan bahasa alami ke dalam query formal dan menghasilkan respon bahasa alami berdasarkan hasil dari database pencarian.

- **Komponen Database**

Ini merupakan fungsi tradisional dari *Database Management*. Sebuah leksikon adalah tabel yang digunakan untuk memetakan kata-kata input alami ke objek formal (nama relasi, atribut nama, dll) dari database. Kedua parser dan juru semantik memanfaatkan leksikon. Sebuah generator bahasa alami mengambil tanggapan resmi sebagai masukan, dan memeriksa pohon parsing untuk menghasilkan respon bahasa alami yang memadai. sistem database bahasa alami memanfaatkan pengetahuan sintaksis dan pengetahuan tentang database sebenarnya dalam rangka untuk benar berhubungan masukan bahasa alami dengan struktur dan isi dari database tersebut. pengetahuan sintaksis biasanya berada dalam komponen linguistik dari sistem, khususnya dalam analisa sintaks sedangkan pengetahuan tentang database sebenarnya berada sampai batas tertentu dalam model data semantik digunakan. Pertanyaan yang dimasukan dalam bahasa alami diterjemahkan ke dalam sebuah pernyataan dalam bahasa query formal. Setelah pernyataan itu jelas terbentuk, query diproses oleh sistem manajemen database untuk menghasilkan data yang dibutuhkan. Data ini kemudian diteruskan kembali ke komponen bahasa alami di mana rutinitas generasi memproduksi versi bahasa permukaan respon.

### 2.6.2 Tahapan NLIDB

- **Scanner (*Tokenizing*)**

Proses scanning bertujuan mengelompokkan masukan kalimat ke dalam token. Scanner mengubah kalimat menjadi daftar kata yang

tergolong token beserta data dan membuang kata-kata yang dapat diabaikan. Langkah – langkah proses scanning adalah sebagai berikut:

1. Membaca masukan kalimat
2. Mengubah kalimat ke dalam bentuk list kata – kata
3. Dari list kata tersebut akan dibuang karakter atau kata yang tidak berarti (dapat diabaikan).

- **Analisis Sintak (*Parser*)**

Parser melakukan pelacakan masukan kalimat untuk mendapatkan langkah pembentukannya. Langkah–langkah ini dilakukan dari atas ke bawah, yaitu dari simbol awal sampai ke kalimat yang dihasilkan. Parser menganalisis sintaks daftar kata hasil scanning sesuai dengan aturan produksi yang ditentukan. Langkah–langkah dalam proses scanning adalah sebagai berikut:

1. Membaca daftar kata hasil scanning yang sudah terseleksi
2. Menentukan frase atribut
3. Memeriksa apakah ada ekor atribut pada frase atribut
4. Menentukan frase kondisi

Teknik yang digunakan dalam parsing adalah perbedaan list. Perbedaan list awal dengan ekor list akan menyisakan satu kata, kemudian kata tersebut diperiksa posisinya dalam query sesuai dengan aturan produksi untuk menentukan apakah termasuk frase atribut atau frase kondisi, apakah atribut atau konstanta.

- **Analisis Semantik (*Translator*)**

Translator berfungsi untuk mengubah pohon sintak hasil parsing ke tipe – tipe query yang sesuai. Langkah – langkah proses penerjemahan dalam translator adalah sebagai berikut:

1. Membaca pohon sintaks hasil parsing.
2. Menempatkan atribut dari frase atribut ke posisi atribut pertama dalam notasi tipe query.
3. Menempatkan atribut dan konstanta (data, int, opr) dari frase kondisi ke dalam notasi tipe query sesuai posisinya.

Aturan produksi yang ditentukan dapat menghasilkan kalimat yang kompleks, karena frase atribut dapat memiliki satu ekor atribut dan frase kondisi dapat mempunyai nol atau satu ekor frase kondisi. Akan tetapi, memperhatikan pertanyaan yang biasa digunakan untuk mengakses data dari basis data akademik, maka translator hanya dirancang untuk mengubah pohon sintaks yang susunan frase atribut dan frase kondisinya sesuai dengan pola pertanyaan – pertanyaan.

- **Analisis Pragmatik (*Evaluator*)**

Evaluator berfungsi menentukan jawaban akhir query berdasarkan hasil keluaran translator yang berupa penggolongan query menurut tipenya dengan urutan atribut sesuai dengan notasi masing – masing tipe query. Proses yang terjadi dalam evaluator adalah pencarian relasi yang terdapat dalam query dan pencarian nilai yang tepat dalam basis data sesuai dengan relasi tersebut. Setiap tipe query mempunyai urutan atribut yang berbeda yang akan menentukan proses evaluasinya.

- **Aturan Produksi**

Bahasa Indonesia telah memiliki grammar dan aturan produksi namun dalam penggunaannya untuk melakukan query basis data perlu diberikan aturan produksi secara khusus (Hartati dan Zuliarso, 2011). Pola pertanyaan yang diberikan dalam bahasa alami untuk mendapatkan hasil dari query sistem basis data perlu disusun ke dalam suatu aturan produksi. Aturan produksi dibangun dengan melakukan pemetaan keteraturan pola dari suatu pertanyaan.

### 2.6.3 Keuntungan NLIDB

- Tidak perlu memahami bahasa pemrograman

Salah satu keuntungan dari NLIDB seharusnya bahwa pengguna tidak diperlukan untuk belajar bahasa komunikasi buatan. bahasa query formal seperti SQL sulit untuk belajar dan menguasai, setidaknya oleh non-komputer-spesialis.

- Sederhana, mudah digunakan

Pertimbangkan database dengan bahasa query atau bentuk tertentu dirancang untuk menampilkan query. Sementara sistem NLIDB hanya membutuhkan input tunggal, bentuk berbasis mungkin berisi beberapa masukan (bidang, kotak scroll, kotak combo, radio tombol, dll) tergantung pada kemampuan formulir. Dalam hal bahasa query, pertanyaan mungkin perlu diungkapkan menggunakan beberapa pernyataan yang mengandung satu atau lebih pertanyaan sub dengan beberapa operasi bersama sebagai konektor. Saat ini semua sistem NLIDB hanya dapat menangani beberapa subset dari bahasa alami dan tidak mudah untuk menentukan subset ini. Bahkan beberapa sistem NLIDB tidak bisa menjawab pertanyaan-pertanyaan tertentu milik subset mereka sendiri. Hal ini tidak terjadi di bahasa formal. Cakupan bahasa formal jelas dan pernyataan yang mengikuti aturan yang diberikan dijamin untuk memberikan jawaban yang sesuai.

- Baik untuk Beberapa Pertanyaan

Telah berpendapat bahwa ada beberapa jenis pertanyaan (mis pertanyaan yang melibatkan negasi, atau kuantifikasi) yang dapat dengan mudah dinyatakan dalam bahasa alami, tapi itu tampaknya sulit (atau setidaknya membosankan) untuk mengekspresikan menggunakan antarmuka grafis atau bentuk berbasis. Misalnya, "Yang departemen tidak memiliki programmer?" (Negasi), atau "Yang perusahaan memasok setiap departemen?" (Universal kuantifikasi), dapat dengan mudah dinyatakan dalam bahasa alami, tetapi mereka akan sulit untuk mengungkapkan dalam banyak grafis atau bentuk- antarmuka berbasis. Pertanyaan seperti di atas dapat, tentu saja, dinyatakan dalam bahasa query database seperti SQL, tapi ekspresi bahasa query database yang kompleks mungkin harus ditulis.

- Toleransi kesalahan

Sebagian besar sistem NLIDB memberikan beberapa toleransi untuk kesalahan tata bahasa ringan, sementara di sistem komputer; sebagian besar waktu, leksikon harus persis sama seperti yang didefinisikan, sintaks harus benar mengikuti aturan-aturan tertentu, dan setiap kesalahan akan menyebabkan input secara otomatis ditolak oleh sistem. Dalam kasus



kalimat tidak lengkap, sebagian besar computer sistem tidak memberikan dukungan apapun.

- Mudah Digunakan untuk Beberapa Tabel Database

Pertanyaan yang melibatkan beberapa tabel database seperti "daftar alamat petani yang mendapat bonus lebih besar dari 10.000 rupee untuk tanaman gandum", sulit untuk membentuk di antarmuka pengguna grafis dibandingkan dengan antarmuka bahasa alami

## 2.7 Receiver Operating Characteristic (ROC)

Kurva ROC pertama kali digunakan pada perang dunia II untuk menganalisis sinyal radar sebelum dikembangkan dalam signal detection theory. Berdasarkan serangan di Pearl Harbon tahun 1941, tentara Amerika melakukan riset untuk meningkatkan ketepatan prediksi dalam mendeteksi sinyal radar pesawat Jepang.

Kurva ROC adalah tool dua dimensi yang digunakan untuk menilai kinerja klasifikasi yang menggunakan dua class keputusan, masing-masing objek dipetakan ke salah satu elemen dari himpunan pasangan, positif atau negatif. Pada kurva ROC, TP rate diplot pada sumbu Y dan FP rate diplot pada sumbu X. Untuk klasifikasi data mining, nilai AUC dapat dibagi menjadi beberapa kelompok (Gorunescu, 2011).

- 0.90-1.00 = *Excellent Classification*
- 0.80-0.90 = *Good Classification*
- 0.70-0.80 = *Fair Classification*
- 0.60-0.70 = *Poor Classification*
- 0.50-0.60 = *Failure*

Kurva ini banyak digunakan untuk menilai hasil prediksi, kurva ROC adalah teknik untuk memvisualisasikan, mengatur, dan memilih pengklasifikasian berdasarkan kinerja mereka (Gorunescu, 2011). Penggunaan kurva ROC semakin populer dalam berbagai aplikasi terutama dalam bidang medis, radiologi, dan processing image. *Receiver Operating Characteristic* (ROC) adalah hasil pengukuran klasifikasi dalam bentuk 2-dimensi. Berikut ada empat peluang yang dapat diformulasikan dalam tabel kontingensi 2 x 2 untuk menganalisis ROC

**Tabel 2.2** Kontingensi ROC

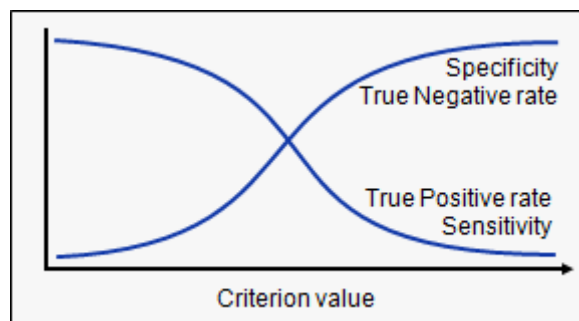
		Kelas Sebenarnya	
		Benar	Salah
Kelas Prediksi	Positif	Benar Positif	Salah Positif
	Negatif	Benar Negatif	Salah Negatif

Adapun kriteria ROC adalah sebagai berikut:

- True Positive Rate disebut juga Sensitivity (TPR)= $TP/(TP+FN)$
- True Negative Rate disebut juga Specifity (TNR)=  $TN/(TN+FP)$
- Accuracy =  $(TP+TN)/(TP+FP+TN+FN)$ .

Dimana:

- TP = True Positive yaitu klasifikasi yang dari kelas yang positif
- FN = False Negative yaitu kesalahan Type II
- FP = False Positive atau kesalahan Type I



**Gambar 2.4** Kriteria ROC (MedCalc Software bvba, 2010)

Jika nilai kriteria yang dipilih lebih tinggi, maka bagian FP akan menurun dan specifity akan meningkat, namun TP dan sensitivity akan menurun. Sebaliknya jika nilai criteria yang dipilih lebih rendah, maka bagian TP akan meningkat, namun bagian TN dan specificity akan menurun (MedCalc Software bvba, 2010).

Area Under Curva (AUC) adalah luas daerah di bawah kurva ROC, bila nilainya mendekati satu, maka model yang didapat lebih akurat. Berdasarkan gambar diatas maka dapat dilihat karakteristik dari AUC adalah sebagai berikut:

- Area maksimum adalah 1
- Jika ROC = 0,5 maka model yang dihasilkan belum terlihat optimal
- Sedangkan jika ROC > 0,5 maka model yang dihasilkan akan lebih baik

Formula AUC (Brefeld, 2005):

$$AUC = \frac{\sum_{i=0}^n \sum_{j=1}^n 1f(x_i^+)f(x_j^-)}{n^+n^-} \dots\dots\dots(2.1)$$

Keterangan:

- $f(.)$  = nilai suatu fungsi
- $x^+$  dan  $x^-$  = sampel positif dan negatif
- $n^+$  dan  $n^-$  = jumlah sampel positif dan negative

(MedCalc Software bvba, 2010)



*Halaman ini sengaja dikosongkan*

## BAB III

### METODE PENELITIAN

#### 3.1 Metode Penelitian

Pada bab ini akan diuraikan langkah-langkah penelitian tesis ini dan penjelasan dari masing-masing langkah penelitian tersebut. Langkah-langkah tersebut dapat dilihat pada gambar dibawah ini:



**Gambar 3.1** Alur Metode Penelitian

#### 3.2 Penyiapan data

penyiapan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam rangka mencapai tujuan penelitian. Data yang digunakan dalam penelitian ini didapatkan sebuah Sistem Informasi Manajemen Rumah sakit yang terdapat di kota Surabaya. Dalam tahapan penyiapan ini terdiri dari tiga aktivitas yaitu pengambilan data, pengumpulan data serta penentuan proses bisnis dan aktivitas tiap sistm.

Data yang diambil berupa aktivitas pengguna dalam proses bisnis farmasi. Data tersebut selanjutnya akan digunakan sebagai acuan dalam keakuratan hasil ekstraksi pada penelitian ini. Data tersebut sesuai dengan standar pada *event log*. Seperti yang dijelaskan pada bab sebelumnya bahwa komponen dalam sebuah *event log* yang dibutuhkan dalam *process mining* yaitu *case ID*, *task*, *originator*, *timestamp*.

### 3.3 Pembuatan bahasa alami yang terstruktur

Sebelum masuk ke dalam tahapan proses NLIDB langkah awal yang perlu dilakukan yaitu memetakan masukan yang akan ditangani di dalam aplikasi. Masukan dalam hal ini adalah kalimat perintah. Contoh kalimat yang akan ditangani di dalam aplikasi diantaranya:

1. Tampilkan <Proses Bisnis>

**Keterangan :** menampilkan *event logs* untuk semua aktivitas dalam sebuah proses bisnis

2. Tampilkan <Aktivitas>

**Keterangan :** menampilkan aktivitas tertentu didalam proses bisnis

3. Tampilkan <Aktivitas 1> <Aktivitas 2> .... <Aktivitas n>

**Keterangan :** menampilkan *event logs* untuk beberapa aktivitas sekaligus

4. Tampilkan <Aktivitas/Proses Bisnis> <Bulan> <Tahun>

**Keterangan :** menampilkan *event logs* untuk aktivitas/proses bisnis tertentu pada bulan dan tahun yang ditentukan

5. Tampilkan <Aktivitas 1> <Aktivitas 2> ... <Aktivitas n> <Bulan> <Tahun>

**Keterangan :** menampilkan *event logs* untuk beberapa aktivitas sekaligus pada bulan dan tahun yang ditentukan

6. Tampilkan <Aktivitas/Proses Bisnis> dari <Tanggal 1> sampai <Tanggal 2>

**Keterangan :** menampilkan *event logs* untuk aktivitas/proses bisnis tertentu untuk dengan range tanggal yang ditentukan sebelumnya

7. Tampilkan <Aktivitas/Proses Bisnis> urutkan <Atribut *event logs*>

**Keterangan :** menampilkan *event logs* untuk aktivitas/proses bisnis tertentu untuk kemudian diurutkan berdasarkan atribut dalam event logs

8. Tampilkan <Aktivitas/Proses Bisnis> saring <Atribut *event logs*> <Nilai atribut>

**Keterangan :** menampilkan *event logs* untuk aktivitas/proses bisnis tertentu untuk kemudian di filter berdasarkan atribut tertentu

9. Tampilkan <Aktivitas/Proses Bisnis> saring <Atribut *event logs*>  
<Nilai atribut> urutkan <Atribut *event logs*>

**Keterangan :** menampilkan *event logs* untuk aktivitas/proses bisnis tertentu untuk kemudian di filter berdasarkan atribut tertentu

Dari kalimat perintah tersebut maka perlu dipetakan menurut sebuah pola kalimat yang baku agar dapat diproses ke dalam komputer. Bahasa Indonesia telah memiliki grammar dan aturan produksi namun dalam penggunaannya untuk melakukan query ke dalam basis data perlu diberikan aturan secara khusus (Hartati dan Zuliarso, 2011) . Pola kalimat perintah menjadi masukan dalam penelitian ini dapat disusun dengan notasi aturan *Backus Naur Form* (Backus dkk., 1963) menjadi seperti berikut

- a.  $S \rightarrow \langle \text{Perintah} \rangle$
- b.  $\langle \text{Perintah} \rangle \rightarrow \langle \text{Tampilkan, Saring, Urutkan} \rangle$
- c.  $\langle \text{Atribut} \rangle \rightarrow \langle \text{Nama Proses Bisnis} \rangle \mid \langle \text{Semua Aktivitas} \rangle$
- d.  $\langle \text{Objek} \rangle \rightarrow \langle \text{Atribut } event logs \rangle$
- e.  $\langle \text{Keterangan Objek} \rangle \rightarrow \langle \text{Tanggal} \rangle \mid \langle \text{Bulan} \rangle \mid \langle \text{Tahun} \rangle \mid \langle \text{Nilai Atribut} \rangle$
- f.  $\langle \text{Nama Proses Bisnis} \rangle \rightarrow \langle \text{Ditentukan berdasarkan sistem dan proses yang ditentukan} \rangle$
- g.  $\langle \text{Nama Aktivitas} \rangle \rightarrow \langle \text{Ditentukan berdasarkan proses yang ditentukan} \rangle$
- h.  $\langle \text{Tanggal} \rangle \rightarrow \langle 1 \text{ s/d } 31 \rangle$
- i.  $\langle \text{Bulan} \rangle \rightarrow \langle \text{Januari s/d Desember, Ini} \rangle$
- j.  $\langle \text{Tahun} \rangle \rightarrow \langle 2016, 4 \text{ digit angka Tahun} \rangle$
- k.  $\langle \text{Nilai Atribut } event logs \rangle \rightarrow \langle \text{Mengandung karakter, angka, kombinasi karakter, nama orang, waktu} \rangle$

Aplikasi pada penelitian ini hanya dapat menerima masukan bahasa alami berdasarkan aturan produksi yang sudah dibuat.

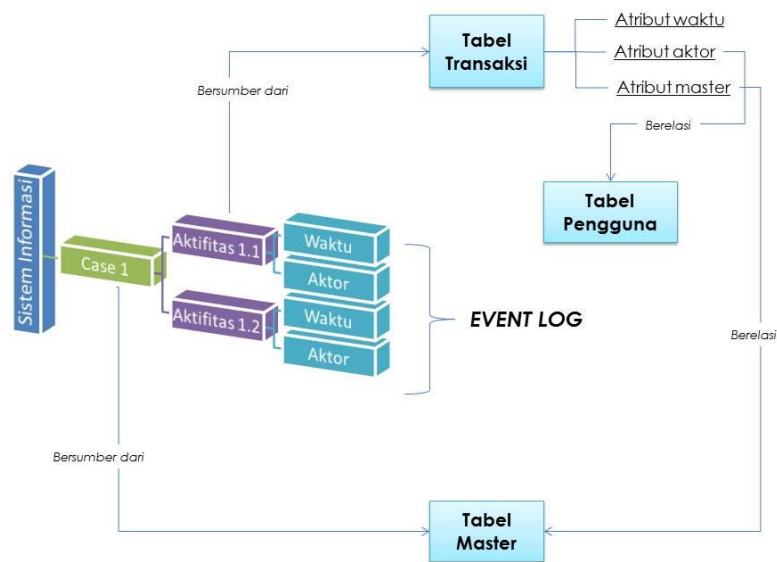
### 3.4 Mendefinisikan informasi *event logs* dalam basis data

*Process mining* bertujuan untuk menemukan, memantau dan memperbaiki proses nyata dengan mengekstrak pengetahuan dari log peristiwa yang tersedia di



sistem informasi. Beberapa teknik *process mining* muncul dan hasilnya telah terbukti dalam berbagai studi kasus. Biasanya, *event logs* sederhana berfungsi sebagai titik awal untuk *process mining* (Calvanese dkk., 2015). *Event log* merupakan sebuah aktivitas yang sudah dilakukan oleh aktor dalam sistem informasi dalam satu waktu, kemudian diurutkan serta dilacak dalam keseluruhan proses yang disebut *traces*. Jadi untuk melakukan proses ekstraksi seorang analis bisnis perlu menentukan sebuah *case* dalam sebuah sistem informasi yang didalamnya mengandung aktifitas. Contoh *case* pada sebuah sistem informasi manajemen rumah sakit yaitu distribusi barang pada farmasi. Aktifitas yang terdapat dalam *case* tersebut dimulai dari pembelian barang, permintaan barang dari unit, persetujuan permintaan barang ke unit sampai dengan pemakaian barang ke pasien. Pada contoh *case* distribusi barang farmasi tersebut dapat dijabarkan atribut *Case ID* pada sebuah *event log* adalah nama barang tersebut, atribut *TaskName* adalah aktifitas yang terjadi, atribut *Timestamp* merupakan waktu penanda aktifitas tersebut dilakukan, atribut *Originator* merupakan aktor yang melakukan aktifitas tersebut ( i J. C. A. M. Buijs, 2010) .

Dalam sistem informasi atribut *event log* tersebut tersimpan dalam database sistem informasi. Seperti pada contoh sebelumnya informasi nama barang tersimpan dalam tabel *master\_barang* dengan kolom *nama\_barang*, kemudian tiap aktifitas tercatat dalam satu tabel transaksi, tiap tabel transaksi tersebut juga menyimpan atribut waktu serta pelaku aktifitas dengan kolom yang berbeda. Contoh pada aktifitas pembelian barang dalam database disimpan dalam tabel transaksi *pembelian\_barang* yang memiliki relasi dengan tabel *master\_barang*. Selain itu aktor yang menjalankan tiap aktifitas yang tersimpan dalam tabel transaksi memiliki relasi dengan tabel lain yang biasanya disebut tabel *pengguna*. Dari penjelasan tersebut maka konsep pemetaan ekstraksi event log dapat dilihat pada gambar 3.2



**Gambar 3.2** Pemetaan Ekstraksi event log dalam database

### 3.5 Pemrosesan bahasa alami ke dalam bahasa *SQL*

Dalam penelitian ini bahasa alami dijadikan sebagai masukan untuk melakukan perintah pengambilan data *event logs* yang tersimpan dalam basis data. Pada umumnya untuk mengambil informasi dalam sebuah basis data menggunakan bahasa *Structured Query Language* (SQL). Dalam pemrosesan bahasa alami yang terstruktur sampai menjadi sebuah bahasa SQL yang dipahami oleh komputer perlu dilakukan beberapa tahapan. Tahapan tersebut digunakan agar bahasa alami lebih mudah dikenali sehingga dapat menghasilkan algoritma tertentu yang akan diformulasikan ke dalam bentuk kode sumber. Beberapa tahapan tersebut diantaranya sebagai berikut:

#### 3.5.1 Scanner (*Tokenizing*)

Pemrosesan awal pertanyaan dilakukan oleh scanner yang akan membuat pertanyaan menjadi sebuah daftar token. Pada penelitian ini scanner melakukan 3 aktivitas yakni: menghilangkan tanda baca, memecah pertanyaan dan mengubah menjadi huruf kecil. Potongan program dalam scanner yang digunakan penelitian ini adalah

- a. Membersihkan kalimat tanya dari tanda baca, menggunakan script *string replace* karakter `[]_^,:;(){}&=>*~%?!`  menjadi karakter kosong
- b. Memecah kalimat menjadi daftar kata dan mencacah kata,
- c. Mengubah ke dalam huruf kecil,

### 3.5.2 Analisis Sintak (*Parser*)

Daftar token yang terbentuk akan dianalisa oleh parser apakah sesuai dengan dengan pembentukan pola kalimat yang telah ditetapkan. Penentuan struktur kalimat bisa jadi merupakan pekerjaan yang sulit bergantung dari bahasa yang digunakan. Parsing morfologi merupakan kegiatan yang dilakukan untuk menentukan struktur kata. Pada penelitian ini secara umum parser akan melakukan

- a. Membaca daftar token
- b. Membaca cacah token
- c. Membaca isi token
- d. Membandingkan dengan aturan produksi

Analisis sintaks melakukan pelacakan terhadap token-token yang dihasilkan oleh scanner kemudian dibandingkan dengan daftar token yang tersedia. Jika sesuai dengan daftar token yang ada maka dilihat apakah terdapat kecocokan dengan aturan produksi yang ada.

### 3.5.3 Analisis Semantik (*Translator*)

Translator berfungsi untuk memetakan hasil parser yang sesuai dengan aturan produksi ke dalam bahasa hasil yakni bahasa query, sehingga yang dilakukan pada analisis semantik meliputi:

- a. Membaca hasil scanner
- b. Memetakan ke dalam bahasa Query

Pada penelitian ini token yang ada dalam terminal atribut, **obyek** dan **keterangan obyek** memegang peranan penting dalam translator. Sebagai contoh pertanyaan ” Tampilkan data permintaan barang Amoxilin kepada logistik”, maka dalam translator dapat diambil kata *permintaan barang Amoxilin kepada logistik*, kata yang lain dapat diabaikan, sehingga dari hasil tersebut didapatkan :

- a. **Atribut** nya adalah permintaan barang,
- b. **Obyek** nya adalah barang
- c. **Keterangan obyek** nya adalah Amoxilin dan logistik.

Kalimat tersebut akan ditranslasikan menjadi

```
"SELECT permintaan_barang
FROM event_log
WHERE nama_barang LIKE %amoxilin% and unit=logistik".
```

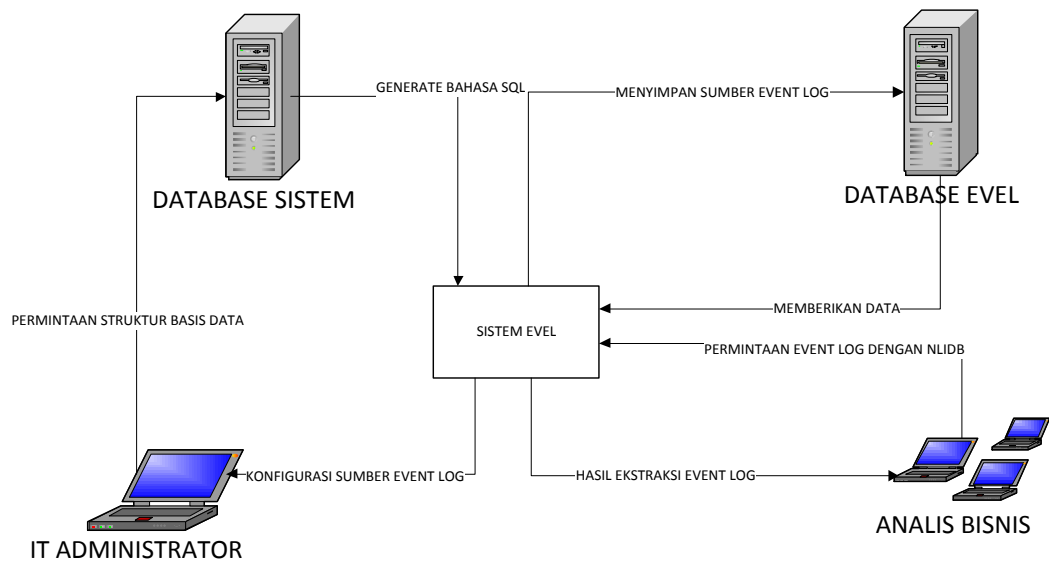
Jadi dalam sistem ini translator hanya akan mengambil token-token yang berguna, sedangkan token yang tidak berguna akan diabaikan. Token yang termasuk dalam token yang diabaikan dalam penelitian ini adalah token dalam <kata sambung> .

### 3.5.4 Analisis Pragmatik (*Evaluator*)

Proses paling akhir dalam tahapan ini adalah memeriksa hasil translasi apakah sudah sesuai dengan kaidah yang dibenarkan. Pada penelitian ini akan memeriksa *token* dari terminal **obyek** dan **keterangan obyek** yang dihasilkan oleh translator. Evaluator akan memeriksa apakah **keterangan obyek** yang dimiliki sesuai dengan **obyek** yang digunakan, sebagai contoh misalkan **obyek** yang dimiliki adalah Barang dan **keterangan obyek** adalah Televisi Sony dan Distributor, maka secara grammar diijinkan namun secara evaluator tidak diperkenankan karena **keterangan obyek** barang dan unit tidak terdapat di dalam daftar *token* yang diperbolehkan

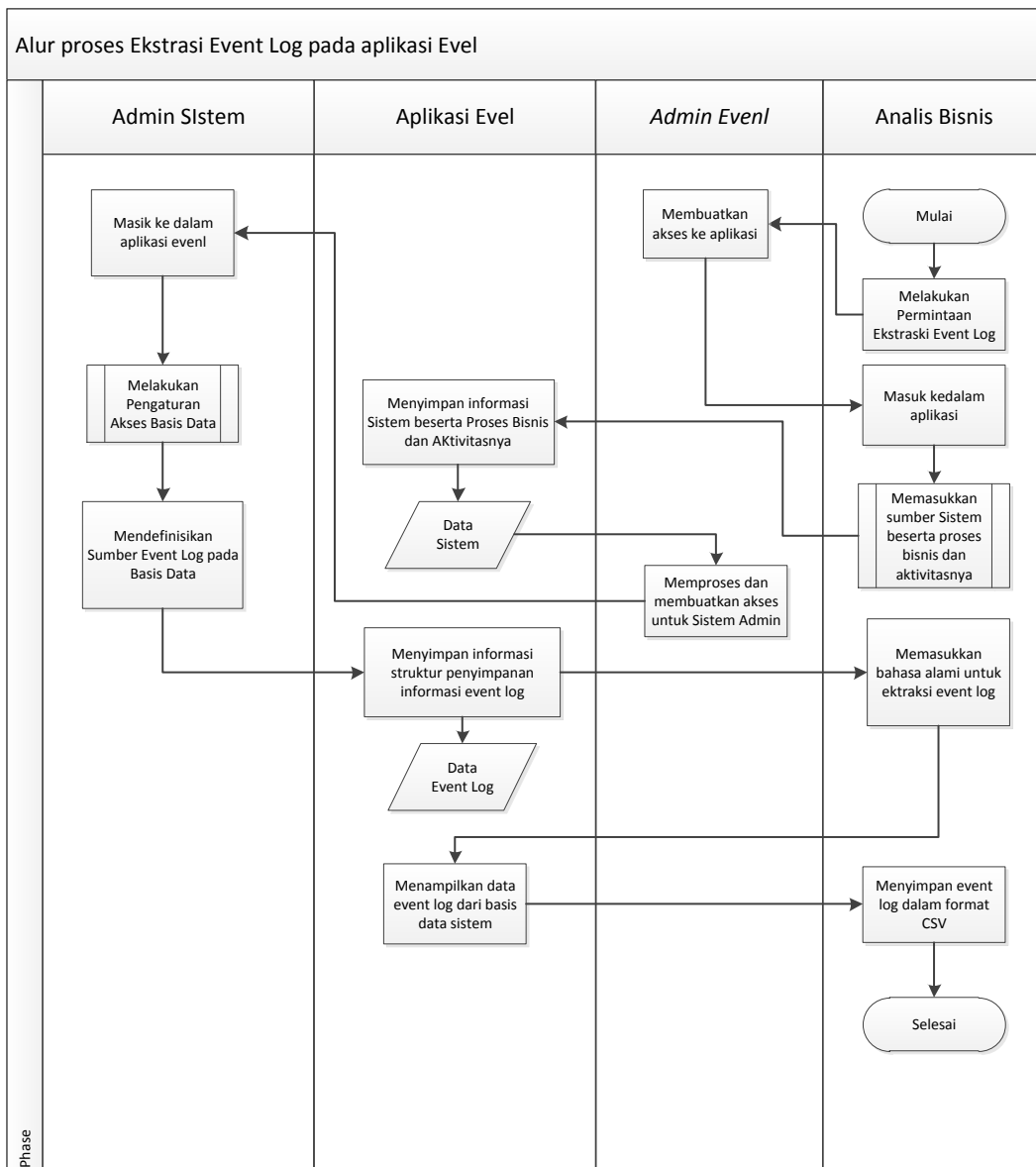
### 3.6 Perancangan desain alat ekstraksi *event log*

Untuk menyelesaikan permasalahan yang sudah dijelaskan pada bagian 1.2 oleh karena itu pada penelitian ini akan menawarkan konsep aplikasi ekstraksi event log berbasis web dengan antarmuka bahasa alami sebagai masukan. Dalam hal ini aplikasi event log yang bernama *Evenl* yang menjadi pen jembatan oleh analis bisnis yang bisa mendapatkan event log, dengan teknologi berbasis web maka memudahkan penggunaan alat ini karena hanya membutuhkan koneksi internet untuk memakainya. Aplikasi ini mengutamakan keamanan basis data sebuah sistem informasi namun juga memberi keluasaan para analis bisnis untuk mendapatkan data event log yang diharapkan. Bentuk desain aplikasi tersebut seperti pada gambar 3.3



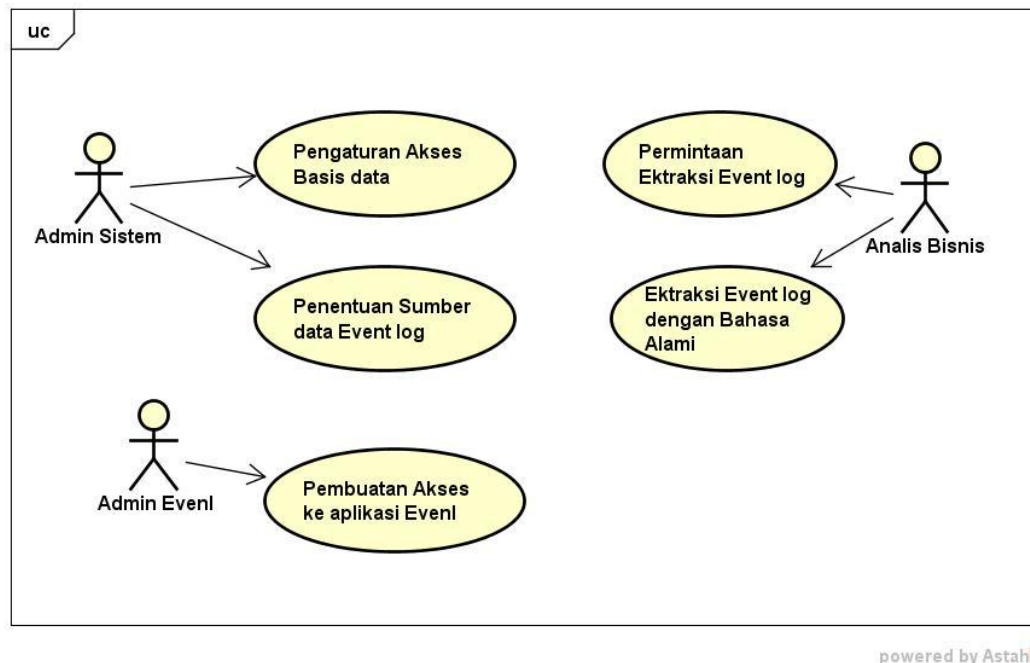
**Gambar 3.3** Konsep desain aplikasi *Eventl*

Dalam konsep aplikasi Eventl diatas aktor pengguna aplikasi dibagi menjadi 3 macam yaitu : 1. Admin *Eventl*, 2. Analis Bisnis, 3. Admin Sistem. Peran utama *Eventl* Admin yaitu mengatur Analis Bisnis dan Sistem Admin agar dapat menggunakan aplikasi *Eventl*, untuk Admin Sistem (*IT Administrator*) memiliki peran untuk memberikan informasi basis data yang mendukung proses ekstraksi dan yang terakhir analis bisnis mengambil hasil ekstraksi *event log* untuk dijadikan kebutuhan process mining yang berguna untuk pemilik sistem. Ketiga macam aktor tersebut akan berkolaborasi dalam aplikasi *Eventl* yang bertujuan untuk ekstraksi *event log*. Untuk lebih jelasnya gambaran proses ekstraksi pada aplikasi Eventl dapat dilihat pada gambar 3.4.



**Gambar 3.4** Alur Proses Ekstraksi pada aplikasi *Eventl*

Dari alur proses ekstraksi pada aplikasi *Eventl* terlihat aktivitas untuk tiap aktor , sehingga untuk memperjelas aktivitas tersebut akan digambarkan dalam bentuk diagram *use case* yang terlihat pada gambar 3.5 dibawah ini



**Gambar 3.5** Use Case aplikasi *Event*

Setelah pendefinisian fungsi untuk tiap aktor selanjutnya akan dibuat desain basis data dan desain antarmuka yang akan dibahas pada bagian pembahasan.

### 3.6.1 Rancangan konversi hasil ke dalam format XES

Tahapan ini bertujuan untuk mengkonversi hasil data yang berasal dari aplikasi ke dalam format event log yang dibutuhkan. ProM Framework adalah aplikasi yang digunakan untuk melakukan analisa dalam bidang proses mining, aplikasi tersebut membutuhkan format data MXML sebagai data sumbernya. Untuk dapat mengkonversi sebuah event log ke dalam format MXML menggunakan aplikasi tambahan yang bernama ProM Import Framework yang membutuhkan beberapa *plugin java* untuk menjalankan proses konversi. Dibutuhkan keahlian dalam domain ini agar bisa berjalan sesuai dengan harapan, hal ini menjadi masalah baru bagi para analis bisnis. Namun dengan munculnya standar event log yang sudah dibuat oleh *Architecture of Information Systems research group* di Eindhoven University of Technology yang berkolaborasi dengan Fluxicon maka permasalahan sebelumnya dapat terpecahkan. Format event log tersebut bernama XES, format event log ini digunakan juga oleh (i J. C. A. M. Buijs, 2010) agar dapat digunakan dalam proses mining. Dengan bantuan

aplikasi yang sudah disediakan <http://www.xes-standard.org> maka hasil event log yang didapat dalam proses NLIDB pada penelitian ini akan dapat dikonversi ke dalam format XES. Selanjutnya event log tersebut dapat digunakan untuk kebutuhan *process mining*.

### 3.7 Pengujian ekstraksi

Skenario dalam pengujian aplikasi ini digunakan untuk memastikan hasil aplikasi pada penelitian ini dapat digunakan pada aplikasi ProM Framework untuk melakukan analisis proses mining. Oleh karena pada tahap pengujian ini hanya fokus terhadap masukan dan hasil keluaran maka jenis pengujian *Black Box* perlu digunakan. Tahapan pada proses pengujian ini yaitu :

1. Pengujian ekstraksi dengan bahasa alami

Pada proses ini akan dimasukkan bahasa alami untuk pengambilan data event logs yang berasal dari sistem informasi. Menguji hasil proses NLIDB dengan data sebenarnya

2. Menguji hasil konversi proses NLIDB ke dalam format XES

Sebelum dapat digunakan dalam aplikasi proses mining seperti Disco atau melalui alat konverter lain seperti pada web <http://www.satcos.in/processmining/index.php> , maka hasil ekstraksi event log yang akan dipakai harus divalidasi sesuai dengan format standar event log.

3. Pengujian data event log XES pada aplikasi process mining

Setelah hasil output sudah sesuai format maka selanjutnya data tersebut akan digunakan pada aplikasi proses mining. Aplikasi yang digunakan untuk menguji format XES yang didapat yaitu ProM dan disco dengan algoritma tertentu.

4. Pengujian akurasi ekstraksi event log

Pengujian ini dilakukan dengan cara menulis sebanyak 100 bahasa alami dengan struktur kalimat perintah ekstraksi yang sudah ditentukan sebelumnya. Di dalam 100 kalimat perintah ekstraksi tersebut akan dibagi beberapa kalimat yang sesuai ataupun kalimat yang tidak sesuai (dalam arti terdapat kesalahan penulisan atau



kesalahan struktur kalimat). Hal tersebut dilakukan karena bahasa alami memiliki lingkup yang luas sehingga tidak menutup kemungkinan sering terjadi kesalahan-kesalahan masukan dari pengguna alat. Sebelum proses pengujian 100 kalimat perintah ekstraksi yang sudah dibuat dimasukan kedalam alat ekstraksi untuk diketahui hasilnya maka perlu dilakukan sebuah tahapan *expert judgement* untuk menilai prediksi hasil oleh analis bisnis dari kalimat perintah ekstraksi tersebut yang kemudian akan disesuaikan dengan hasil sebenarnya.

#### 5. User Acceptance Test (UAT) oleh Analis Bisnis

Tahap akhir dari pengujian ini adalah untuk membuktikan kesesuaian bentuk format *event log* yang ditampilkan dengan perintah bahasa alami yang di masukan sesuai dengan kebutuhan para analis bisnis.

### 3.8 Analisa Hasil

Dari beberapa proses scenario pengujian menggunakan metode *black box* dapat diketahui bahwa peroses ekstraksi pada penelitian layak dapat dilakukan. Selanjutnya pada pengujian konversi digunakan untuk membuktikan hasil konversi dapat dipakai dalam aplikasi *process mining*. Selain itu terdapat pengujian akurasi ekstraksi agar dapat memberikan gambaran tingkat keakuratan dengan membandingkan data prediksi dan data hasil yang sebenarnya akan didapatkan nilai sensifitas (*true positif rate*) dan spesifitas (*true negatif rate*). Kemudian dengan hasil tersebut akan dimasukan kedalam *software* statistik SPSS agar dapat melihat kurva ROC dari kinerja klasifikasi alat ekstraksi ini. Hasil gambaran kurva ROC akan dinilai jika diatas garis *baseline* maka kinerja klasifikasi dapat dikatakan baik. Selanjutnya untuk nilai plot kurva ROC yang akan didapatkan akan dilihat masuk dalam kelompok kinerja klasifikasi *excellent*, *good*, *fair*, *poor* atau *failure*. Apapun hasil yang didapatkan dalam proses pengujian akan dijelaskan lebih detail dan jelas sehingga memberikan sebuah hasil penelitian akurat.

Pengujian terakhir dilakukan untuk menjawab bahwa dengan bahasa alami proses ekstraksi dan konversi *event logs* dapat digunakan oleh analis bisnis tanpa

pengetahuan mengenai bahasa pemrograman dan bahasa SQL. Pembuktian hal tersebut dilakukan dengan pengujian UAT yang melibatkan analis bisnis di lingkup sistem rumah sakit dan Odoo ERP. Proses pengujian ini dilakukan dengan cara memberikan daftar fitur serta hasilnya dalam proses ekstraksi dan konversi dengan bahasa alami sesuai dengan proses bisnis terkait pada sistem yang berjalan

.

*Halaman ini sengaja dikosongkan*



## **BAB IV**

### **PEMBAHASAN**

Bab ini akan memaparkan tahap pengumpulan data, kemudian dari data dikumpulkan tersebut akan dianalisa informasi *event log* yang tersimpan, proses pembuatan bahasa natural untuk kepentingan ekstraksi, konversi bahasa bahasa alami ke dalam syntax SQL, konversi hasil ekstraksi ke dalam format XES dan yang terakhir analisa hasil dengan sebuah perancangan sistem untuk alat ekstraksi dengan bahasa alami yang akan dibangun dalam penelitian ini.

#### **4.1 Penyiapan data**

Tahap penyiapan data merupakan tahapan untuk mempersiapkan data yang digunakan dalam penelitian ini. Proses pada tahapan ini meliputi pengumpulan data, penentuan proses bisnis yang digunakan, kemudian pengambilan informasi *event log* yang tersimpan dalam basis data.

##### **4.1.1 Pengambilan data**

Pengambilan data event logs pada sebuah sistem yang berjalan dilakukan pada instansi Rumah Sakit Universitas Airlangga, pada proses pengambilan tersebut telah dilakukan uji etik (menunjukkan bahwa penelitian layak dan tidak merugikan). Uji etik tersebut dilakukan karena Rumah Sakit menyimpan informasi penting yang berkaitan dengan penyakit maupun informasi pasien. Hal tersebut termasuk dalam data yang tidak boleh disebar luaskan dan dijaga kerahasiaanya. Pada sistem lain yaitu Odoo ERP proses pengambilan data dilakukan dengan simulasi sebuah usaha garment dan melakukan beberapa proses dan aktivitas tertentu sehingga terlihat seperti pada kejadian yang sebenarnya

##### **4.1.2 Pengumpulan data**

Pengumpulan data ditujukan untuk 2 sistem selain SAP yang berbeda yaitu : SIM Rumah Sakit dan Odoo ERP versi 9. Pengumpulan data dilakukan dalam beberapa waktu yang berbeda serta menggunakan teknologi yang berbeda. Pada tabel 4.1 akan diterangkan lebih jelas tentang informasi pengumpulan data dari ketiga sistem tersebut.

**Tabel 4.1** Spesifikasi Sistem yang akan dijadikan ujicoba

Nama Sistem	Teknologi			Waktu	Keterangan
	Platform	Bahasa	DBMS		
SIM Rumah Sakit	Web	PHP	PostgreSQL	September 2017	Berisikan data mulai tahun 2014 sampai tahun 2017
Odoo versi 9	Web	Python	PostgreSQL	Desember 2017	Menggunakan demo data Odoo

Pada penelitian sebelumnya (J. Buijs, 2010; Piessens, 2011) menggunakan sistem SAP untuk ujicoba, SAP menggunakan vendor DB2, Oracle dan Microsoft SQL untuk menjalankan sistemnya. Untuk kebutuhan sistem tidak terlalu besar dan kompleks kebanyakan pengembang sistem memilih menggunakan vendor basis data PostgreSQL. Salah satu alasan penggunaan vendor basis data tersebut karena dapat digunakan secara gratis walaupun mereka juga menyediakan versi berbayar. Oleh karena itu pada penelitian ini akan dikelompokkan data berdasarkan vendor DBMS yang digunakan yaitu PostgreSQL karena hal ini berpengaruh saat proses konversi ke dalam bahasa SQL.

#### **4.1.3 Penentuan proses bisnis dan aktivitas**

Pada setiap sistem yang akan dipilih 1 bisnis proses, penentuan proses pertama digunakan sebagai dasar untuk dasar dalam penyelesaian permasalahan dalam ekstraksi *event log*. Dari SIM Rumah Sakit proses bisnis yang dipilih adalah “Distribusi Barang” kemudian untuk sistem Odoo ERP menggunakan proses bisnis “Order Penjualan”. Setiap sistem memiliki beberapa proses bisnis yang terhubung satu sama lain serta tiap proses bisnis memiliki masukan dan hasil yang berbeda, perbedaan tersebut didasarkan dari tujuan pembuatan sistem tersebut. Pada penelitian ini menggunakan SIM Rumah Sakit dan Odoo ERP, untuk SIM

Rumah Sakit pemilihan proses bisnis “Pemberian E-Resep” dan “Distribusi Barang Farmasi”, sesuai pada definisi *event log* pada setiap proses bisnis memiliki isi atribut yang berbeda tergantung dari tujuan hasil dari proses bisnis tersebut. Untuk membuat sebuah kalimat perintah ekstraksi *event log* pada SIM Rumah Sakit maka perlu diperjelas lagi bahwa tiap *event log* adalah kumpulan *trace* yang terdiri 1 kasus (*CaseID*) dengan urutan kejadian (*Activity*) ditandai dengan keterangan waktu (*Timestamp*) yang dilakukan oleh aktor (*Originator*). Untuk memudahkan proses komputerisasi maka pendefinisian proses bisnis dan aktivitas tidak boleh memiliki sktruktur kata yang sama antara satu dengan yang lain karena dapat menimbulkan ambiguitas makna. Untuk itu perlu ditentukan dari awal proses bisnis dan aktivitas dalam SIM Rumah Sakit berdasarkan SOP yang didapatkan adalah sebagai berikut

- Distribusi Barang
  1. Permintaan Barang
  2. Persetujuan Permintaan
  3. Penerimaan Barang

Untuk sistem Odoo ERP penentuan proses bisnis beserta aktivitasnya didasarkan pada dokumentasi Odoo versi 9

- Order Penjualan
  1. Buat Penawaran
  2. Konfirmasi Penawaran
  3. Buat Tagihan
  4. Bayar Tagihan
  5. Persiapan Pengiriman
  6. Selesai Pengiriman
  7. Selesai Penjualan

#### **4.2 Pembuatan bahasa alami yang terstruktur**

Dalam Bahasa Indonesia telah memiliki grammar dan aturan produksi namun dalam penggunaannya untuk mengkonversi bahasa alami ke dalam *query* basis data perlu diberikan aturan yang terstruktur. Dalam sebuah *event log* terdapat memiliki atribut yang dijadikan dasar dalam pembuatan bahasa alami.

Dalam Bahasa Indonesia untuk membangun sebuah kalimat yang baik dan benar perlu memiliki unsur Subjek (S), Predikat(P), Objek (O), Keterangan (K) dan Pelengkap (P). Untuk menerapkan SPOK dalam ekstraksi event log maka perlu mendefinisikan unsur kalimat untuk setiap atribut *event log*. Penjelasan tiap atribut akan menggunakan sebuah contoh proses sederhana “klaim asuransi” yang akan dijelaskan pada keterangan dibawah ini:

1. *CaseID* mewakili sebagai Objek (O)

Dalam sebuah proses tiap kasus memiliki sebuah identitas yang membedakan antara satu dengan yang lain, kasus tersebut memiliki urutan pengerjaannya tersendiri. Pengurusan klaim asuransi identitas pelanggan menjadi sebuah identitas yang menjadi acuan seorang pegawai asuransi untuk mencairkan klaim pasien tersebut. Penentuan *CaseID* sebagai Objek (O) dikenakan tindakan “mencairkan” yang dilakukan oleh “pegawai asuransi”

2. *Activity* mewakili sebagai Predikat (P)

Tiap proses paling tidak memiliki sebuah aktivitas atau lebih. Untuk proses klaim asuransi beberapa aktivitas yang dilakukan oleh aktor yaitu pasien memberikan berkas, pegawai asuransi memvalidasi berkas, pegawai mencairkan klaim. Dalam *event log* sebuah *activity* merupakan aktivitas yang dilakukan aktor tersebut “memberikan”, “memvalidasi”, “mencairkan” maka atribut *activity* dapat digolongkan menjadi Predikat (P) karena mengandung kegiatan.

3. *Timestamp* mewakili sebagai Keterangan waktu (K)

Tiap kasus dalam sebuah proses memiliki urutan masing-masing yang didefinisikan dalam sebuah waktu kejadian. “Pasien A” memberikan berkas klaim pada pukul 08.30 pagi di hari kamis 4 januari 2018, keesokan harinya “Pasien B” memberikan berkas klaim pada pukul 09.22 pada hari jum’at 5 januari 2018. Waktu kejadian tersebut menjabarkan sebuah keterangan akan sebuah kegiatan yang dilakukan sehingga *timestamp* digolongkan dalam sebuah Keterangan waktu (K),



#### 4. *Originator* mewakili sebagai Subjek (S)

Aktivitas atau kegiatan dalam sebuah proses pasti dilakukan oleh aktor, dalam hal ini bisa seseorang ataupun mesin dalam sebuah sistem. Untuk klaim asuransi oleh seorang pasien telah diproses kemudian petugas asuransi mencairkan dana klaim asuransi pasien tersebut. Atribut *originator* dalam *event log* merupakan aktor yang melakukan sebuah kegiatan atau aktivitas pada kurun waktu tertentu. Dalam sebuah kalimat sesuatu yang melakukan sebuah kegiatan disebut sebagai Subjek (S)

Dari penjelasan atribut *event log* dalam struktur Bahasa Indonesia maka akan dibentuk sebuah kalimat perintah yang akan digunakan sebagai masukan kemudian diproses menghasilkan informasi yang diinginkan. Kalimat perintah umumnya terdiri dari unsur kata perintah diikuti keterangan pelengkap. Keterangan pelengkap dalam sebuah kalimat perintah bisa mengandung unsur Subjek (S), Predikat (P), Objek (O) dan Keterangan (K). Untuk lebih jelasnya dalam tabel 4.2 terdapat beberapa kalimat perintah dengan unsur kata perintah dan keterangan pelengkap.

**Tabel 4.2** Struktur kalimat perintah

Kalimat	Kata Perintah	Keterangan Pelengkap
Ambilkan bajuku budi	Ambilkan	$\frac{\text{bajuku}}{O} \quad \frac{\text{budi}}{S}$
Panggilkan toni	Panggilkan	$\frac{\text{toni}}{S}$
Buka pintu rumah	Buka	$\frac{\text{pintu rumah}}{O}$
Pindahkan buku ini	Pindahkan	$\frac{\text{buku ini}}{O}$
Carikan barangku sekarang	Carikan	$\frac{\text{barangku}}{O} \quad \frac{\text{Sekarang}}{K}$
Tolong isikan dokumen	Tolong	$\frac{\text{isikan}}{P} \quad \frac{\text{dokumen}}{O}$

Kalimat perintah dapat dibuat sebuah perintah sebagai masukan ke dalam komputer yang bertujuan untuk ekstraksi informasi dari sebuah data komputer.

Hasil yang diharapkan dalam ekstraksi tersebut adalah menampilkan sebuah data atau informasi yang disebut *event log* yang tersimpan dalam basis data. Oleh karena itu kata perintah yang cocok dapat digunakan untuk perintah ekstraksi adalah “Tampilkan” kemudian keterangan pelengkap dapat berisi mengenai atribut dari *event log*.

#### 4.2.1 Unsur bahasa alami atribut *event log*

Dalam membentuk bahasa alami yang baku tiap kata dalam sebuah kalimat harus sesuai kaidah KBBI (Kamus Besar Bahasa Indonesia), oleh karena itu perlu mendefinisikan isi tiap atribut dalam *event log* yang akan digunakan sebagai perintah atau masukan kemudian diproses ke dalam komputer. Sebagai contoh pada proses bisnis klaim asuransi *CaseID* bisa berupa ID pasien atau nama pasien, *Activity* berisi gabungan dua kata atau frasa dengan struktur (P-O) yaitu “Pemberian Berkas”, “Validasi Berkas”, “Pembayaran Klaim”, *Timestamp* berisikan waktu aktivitas pukul 09.30 tanggal 4 Juli 2017, *Originator* berisi nama aktor (Nama manusia, Entitas Lain) yang melakukan aktivitas tersebut. Dari penjelasan diatas maka dapat dibuat sebuah standar isi atribut *event log* sesuai kaidah KBBI seperti pada tabel dibawah ini

**Tabel 4.3** Isi atribut event log dalam bahasa alami

Atribut <i>event log</i>	Unsur Bahasa Alami	Keterangan
<i>CaseID</i>	Kata benda	Nomer ID, Nama Orang
<i>Activity</i>	Frasa atau gabungan kata kerja dan kata benda	Nama aktivitas diikuti dengan objek
<i>Timestamp</i>	Kata keterangan (Waktu)	Periode, Tanggal, Waktu
<i>Originator</i>	Kata benda	Nama Orang, Nama Entitas

#### 4.2.2 Kalimat perintah ekstraksi

Dari proses penentuan proses bisnis dan aktivitasnya selanjutnya dalam perancangan hasil dari perintah ekstraksi ini yaitu sebuah informasi *event log* yang utuh untuk tiap proses bisnis. Sesuai dari penjelasan sebelumnya mengenai

struktur kalimat perintah yang terdiri dari kata perintah dan keterangan pelengkap, untuk dapat menghasilkan sebuah informasi *event log* yang utuh akan ditambahkan unsur proses bisnis yang diidentifikasi sebagai unsur objek yang terdiri dari beberapa atribut *event log*. Dari penentuan kata untuk proses bisnis dan aktivitas maka struktur kalimat perintah pada SIM Rumah Sakit dan Odoo ERP didefinisikan menjadi beberapa perintah sesuai dari struktur KPE yang telah ditentukan sebelumnya.

### **1. Kalimat perintah ekstraksi pada SIM Rumah Sakit**

Pada penjelasan sebelumnya telah ditentukan tiap bisnis proses dan aktivitasnya selanjutnya akan disesuaikan dengan struktur KPE yang sudah dibangun. Contoh format perpaduan KPE dengan proses bisnis beserta aktivitasnya pada SIM Rumah Sakit adalah sebagai berikut :

- **Jenis kalimat perintah ekstraksi untuk menampilkan berdasarkan proses bisnis**
  - Tampilkan proses Distribusi Barang
  - Tampilkan proses Distribusi Barang pada tanggal 01 januari 2017 sampai 30 februari 2017
- **Jenis kalimat perintah ekstraksi untuk menampilkan berdasarkan aktivitas**
  - Proses bisnis Distribusi Barang
    - Tampilkan aktivitas Permintaan Barang
    - Tampilkan aktivitas Persetujuan Barang
    - Tampilkan aktivitas Penerimaan Barang
    - Tampilkan aktivitas Permintaan Barang pada tanggal 01 januari 2017 sampai 30 februari 2017
    - Tampilkan aktivitas Persetujuan Permintaan pada tanggal 01 januari 2017 sampai 30 februari 2017
    - Tampilkan aktivitas Penerimaan Barang pada tanggal 01 januari 2017 sampai 30 februari 2017
- **Jenis kalimat perintah ekstraksi untuk menampilkan berdasarkan beberapa aktivitas sekaligus**

- Proses bisnis Distribusi Barang
  - Tampilkan Permintaan Barang, Persetujuan Permintaan, Penerimaan Barang
  - Tampilkan Permintaan Barang, Persetujuan Permintaan, Penerimaan Barang pada tanggal 01 januari 2017 sampai 30 februari 2017

Dari KPE yang disesuskan dengan proses bisnis tersebut terdapat beberapa kata atau frasa yang digaris bawahhi menentukan proses analisis sintak pada konversi bahasa alami ke dalam syntax SQL.

## **2. Kalimat perintah ekstraksi pada Odoo ERP**

Pada penjelasan sebelumnya telah ditentukan tiap bisnis proses dan aktivitasnya selanjutnya akan disesuaikan dengan struktur KPE yang sudah dibangun. Contoh format perpaduan KPE dengan proses bisnis beserta aktivitasnya pada Odoo ERP adalah sebagai berikut :

- **Jenis kalimat perintah untuk menampilkan berdasarkan proses bisnis**
  - Tampilkan proses Order Penjualan
  - Tampilkan proses Order Penjualan pada tanggal 01 januari 2017 sampai 30 februari 2017
- **Jenis kalimat perintah ekstraksi untuk menampilkan berdasarkan aktivitas**
  - Proses bisnis Order Penjualan
    - Tampilkan aktivitas Buat Penawaran
    - Tampilkan aktivitas Konfirmasi Penawaran
    - Tampilkan aktivitas Penjualan Selesai
    - Tampilkan aktivitas Buat Penawaran pada tanggal 01 januari 2017 sampai 30 februari 2017
    - Tampilkan aktivitas Kirim Penawaran pada tanggal 01 januari 2017 sampai 30 februari 2017
    - Tampilkan aktivitas Konfirmasi Penjualan pada tanggal 01 januari 2017 sampai 30 februari 2017
    - Tampilkan aktivitas Penjualan Selesai pada tanggal 01 januari 2017 sampai 30 februari 2017

- **Jenis kalimat perintah ekstraksi untuk menampilkan berdasarkan beberapa aktivitas sekaligus**
  - Proses bisnis Order Penjualan
    - Tampilkan aktivitas Permintaan Barang, Persetujuan Permintaan, Penerimaan Barang
    - Tampilkan aktivitas Permintaan Barang, Persetujuan Permintaan, Penerimaan Barang pada tanggal 01 januari 2017 sampai 30 februari 2017

Dari KPE yang disesuskan dengan proses bisnis tersebut terdapat beberapa kata atau frasa yang digaris bawahi menentukan proses analisis sintak pada konversi bahasa alami ke dalam syntax SQL.

#### 4.2.3 Struktur kalimat perintah ekstraksi

Berdasarkan penjelasan mengenai contoh kalimat perintah disertai dengan pendefinisian unsur bahasa alami pada setiap atribut *event log* maka akan terbentuk beberapa struktur kalimat perintah ekstraksi *event log* dengan menggunakan bahasa alami. Tiap struktur kalimat tersebut pada penelitian kali ini selanjutnya akan disebut KPE (Kalimat Perintah Ekstraksi) disertai nomer yang menandakan jenis struktur KPE yang digunakan. KPE yang dapat digunakan sebagai masukan kali ini dibagi menjadi 3 jenis yaitu berdasarkan proses bisnis, berdasarkan aktivitas dan berdasarkan beberapa aktivitas. Dalam sebuah *event log* aktivitas atau kejadian yang terjadi harusurut sesuai dengan kenyataan yang terjadi. Akan tetapi antara *case* satu dengan yang lain seringkali terjadi *overlapping* waktu dalam aktivitas nya, oleh karena itu perlu tambahan pengurutan hasil ekstraksi berdasarkan atribut *event log*. Selain itu untuk memudahkan dalam melihat hasil *event log* pada *case* atau aktor tertentu maka perlu ditambahkan juga sebuah struktur bahasa alami untuk penyaringan hasil ekstraksi berdasarkan atribut *CaseID* dan *Originator*.

- **KPE 1 (Berdasarkan Proses Bisnis) :**

<i>Tampilkan</i>	<i>Proses</i>	<i>Proses bisnis</i>
<i>Kata perintah</i>	<i>Kata bantu</i>	<i>Frasa dengan pola (P-O)</i>

**Keterangan :** menampilkan semua *event log* untuk proses bisnis tertentu

### KPE 1.1

<u>Tampilkan</u>	<u>Proses</u>	<u>Proses bisnis</u>	<u>tanggal/bulan/tahun</u>
<u>Kata perintah</u>	<u>Kata bantu</u>	<u>Frasa dengan pola (P-O)</u>	<u>Kata bantu</u>
<u>Periode,Tanggal/Waktu</u>			
<u>Kata keterangan waktu</u>			

**Keterangan :** menampilkan semua *event log* untuk proses bisnis tertentu di periode waktu tertentu

- **KPE 1.2 :**

<u>Tampilkan</u>	<u>Proses</u>	<u>Proses bisnis</u>	<u>tanggal/bulan/tahun</u>
<u>Kata perintah</u>	<u>Kata bantu</u>	<u>Frasa dengan pola (P-O)</u>	<u>Kata bantu</u>
<u>Periode,Tanggal/Waktu</u>	<u>Saring</u>	<u>Atribut CaseID/Originator</u>	<u>Nilai Atribut</u>
<u>Kata keterangan waktu</u>	<u>Kata perintah</u>	<u>Kata benda</u>	<u>Kata benda</u>

**Keterangan :** menampilkan semua *event log* untuk proses bisnis tertentu di periode waktu tertentu

- **KPE 1.3 :**

<u>Tampilkan</u>	<u>Proses</u>	<u>Proses bisnis</u>	<u>tanggal/bulan/tahun</u>
<u>Kata perintah</u>	<u>Kata bantu</u>	<u>Frasa dengan pola (P-O)</u>	<u>Kata bantu</u>
<u>Periode,Tanggal/Waktu</u>	<u>Urutkan</u>	<u>Atribut CaseID/Originator</u>	<u>Naik/Turun</u>
<u>Kata keterangan waktu</u>	<u>Kata perintah</u>	<u>Kata benda</u>	<u>Kata benda</u>

**Keterangan :** menampilkan semua *event log* untuk proses bisnis tertentu di periode waktu tertentu

- **KPE 2 (Berdasarkan Aktivitas):**

<u>Tampilkan</u>	<u>Aktivitas</u>	<u>Nama Aktivitas</u>
<u>Kata perintah</u>	<u>Kata bantu</u>	<u>Frasa dengan pola (P-O)</u>

**Keterangan :** menampilkan *event log* untuk aktivitas tertentu dalam proses bisnis

- **KPE 2.1 :**

<u>Tampilkan</u>	<u>Aktivitas</u>	<u>Nama Aktivitas</u>	<u>tanggal/bulan/tahun</u>
<u>Kata perintah</u>	<u>Kata bantu</u>	<u>Frasa dengan pola (P-O)</u>	<u>Kata bantu</u>
<u>Periode,Tanggal/Waktu</u>			
<u>Kata keterangan waktu</u>			

**Keterangan :** menampilkan semua *event log* untuk aktivitas tertentu dalam proses bisnis untuk periode waktu tertentu

- **KPE 2.2 :**

<u>Tampilkan</u>	<u>Aktivitas</u>	<u>Nama Aktivitas</u>	<u>tanggal/bulan/tahun</u>
<u>Kata perintah</u>	<u>Kata bantu</u>	<u>Frasa dengan pola (P-O)</u>	<u>Kata keterangan waktu</u>
<u>Periode,Tanggal/Waktu</u>	<u>Saring</u>	<u>Atribut CaseID/Originator</u>	<u>Nilai Atribut</u>
<u>Kata keterangan waktu</u>	<u>Kata perintah</u>	<u>Kata benda</u>	<u>Kata benda</u>

**Keterangan** : menampilkan semua *event log* untuk proses bisnis tertentu di periode waktu tertentu

- **KPE 2.3**

<u>Tampilkan</u> Kata perintah	<u>Aktivitas</u> Kata bantu	<u>Nama Aktivitas</u> Frasa dengan pola (P-O)	<u>tanggal/bulan/tahun</u> Kata bantu
<u>Periode,Tanggal/Waktu</u> Kata keterangan waktu	<u>Urutkan</u> Kata perintah	<u>Atribut CaseID/Originator</u> Kata benda	<u>Naik/Turun</u> Kata benda

**Keterangan** : menampilkan semua *event log* untuk proses bisnis tertentu di periode waktu tertentu

- **KPE 3** (Berdasarkan beberapa Aktivitas sekaligus) :

<u>Tampilkan</u> Kata perintah	<u>Aktivitas</u> Kata bantu	$\left\{ \frac{\text{Nama Aktivitas}}{\text{Frasa dengan pola (P-O)}} \dots n \right\}$
-----------------------------------	--------------------------------	-----------------------------------------------------------------------------------------

**Keterangan** : menampilkan semua *event log* untuk beberapa aktivitas dalam proses bisnis

- **KPE 3.1 :**

<u>Tampilkan</u> Kata perintah	<u>Aktivitas</u> Kata bantu	$\left\{ \frac{\text{Nama Aktivitas}}{\text{Frasa dengan pola (P-O)}} \dots n \right\}$	<u>tanggal/bulan/tahun</u> Kata bantu
<u>Periode,Tanggal/Waktu</u> Kata keterangan waktu			

**Keterangan** : menampilkan semua *event log* untuk beberapa aktivitas dalam proses bisnis untuk periode waktu tertentu

- **KPE 3.2 :**

<u>Tampilkan</u> Kata perintah	<u>Aktivitas</u> Kata bantu	$\left\{ \frac{\text{Nama Aktivitas}}{\text{Frasa dengan pola (P-O)}} \dots n \right\}$	<u>tanggal/bulan/tahun</u> Kata keterangan waktu
<u>Periode,Tanggal/Waktu</u> Kata keterangan waktu	<u>Saring</u> Kata perintah	<u>Atribut CaseID/Originator</u> Kata benda	<u>Nilai Atribut</u> Kata benda

**Keterangan** : menampilkan semua *event log* untuk proses bisnis tertentu di periode waktu tertentu

- **KPE 3.3 :**

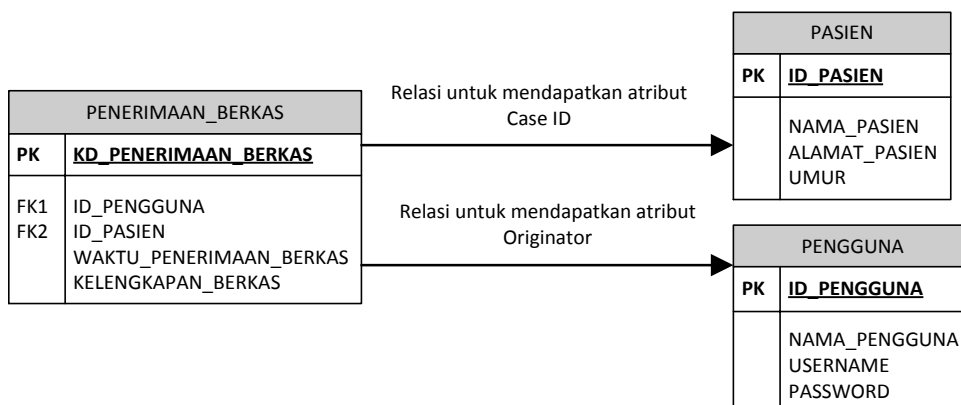
<u>Tampilkan</u> Kata perintah	<u>Aktivitas</u> Kata bantu	$\left\{ \frac{\text{Nama Aktivitas}}{\text{Frasa dengan pola (P-O)}} \dots n \right\}$	<u>tanggal/bulan/tahun</u> Kata bantu
<u>Periode,Tanggal/Waktu</u> Kata keterangan waktu	<u>Urutkan</u> Kata perintah	<u>Atribut CaseID/Originator</u> Kata benda	<u>Naik/Turun</u> Kata benda

**Keterangan** : menampilkan semua *event log* untuk proses bisnis tertentu di periode waktu tertentu

Beberapa jenis KPE yang sudah ditentukan tersebut akan dijadikan sebagai dasar dalam pengujian ekstraksi *event log* bahasa alami dari basis data, pengujian tersebut dilakukan satu-persatu untuk semua sistem dengan proses bisnis beserta aktivitasnya masing-masing.

### 4.3 Mendefinisikan informasi event logs dalam basis data

Dari proses bisnis beserta aktivitas yang sudah ditentukan sebelumnya perlu dicari sumber informasi *event log* yang tersimpan dalam basis data tiap sistem. Pada umumnya tiap aktivitas dalam sebuah proses bisnis terimpan dalam satu tabel basis data, akan tetapi untuk beberapa kasus informasi *event log* untuk tiap aktivitas tersebut bisa jadi tersimpan dalam beberapa tabel yang berelasi satu sama lain. Contoh untuk proses bisnis klaim asuransi, untuk aktivitas pemberian berkas tersimpan dalam satu tabel transaksi “penerimaan\_berkas” namun untuk mendapatkan informasi atribut *CaseID*, *Originator* dalam *event log* tersimpan dalam tabel lain yang memiliki relasi dalam tabel transaksi tersebut. Untuk atribut *CaseID* pada proses bisnis klaim asuransi berisi “Nama Pasien” atau “ID Pasien”, tersimpan dalam tabel tersendiri yaitu tabel “pasien” kemudian untuk *originator* adalah aktor dalam sebuah sistem yang tersimpan pada tabel “pengguna”. Nama tabel untuk tiap sistem bisa berbeda-beda bergantung pada pengembang sistem masing-masing. Contoh hubungan atau relasi antara tabel “penerimaan\_berkas”, tabel “pasien” dan tabel “pengguna” dapat digambarkan seperti gambar 4.1 di bawah ini.



**Gambar 4.1** Contoh Bentuk relasi atribut *event log* dalam desain basis data



Pada contoh diatas menggambarkan informasi atribut *event log* yang tersimpan dalam basis data untuk contoh proses bisnis klaim asuransi pada aktivitas “Penerimaan Berkas” oleh pasien. Dari informasi tersebut akan bisa dibuat sintak SQL untuk tiap proses bisnis beserta aktivitasnya pada setiap sistem pada penelitian ini. Bentuk data SQL yang ditampilkan berbentuk tabel sesuai dengan format *event log* dengan kolom *CaseID*, *Activity*, *Timestamp*, *Originator* yang mendefinisikan atribut pada *event log*.

#### 4.3.1 Sumber tabel dan kolom

Selanjutnya untuk sistem SIM Rumah Sakit dan Odoo ERP. pada tiap sistem proses bisnis beserta aktivitasnya masing-masing akan ditentukan tabel dan kolom yang dijadikan sumber informasi dalam konversi ke dalam syntax SQL untuk tiap atribut event log akan dijelaskan dibawah ini

##### 1. Atribut *event log* pada basis data SIM Rumah Sakit

- Proses Bisnis **Distribusi Barang**

*CaseID* yang dipakai ID Permintaan Barang dalam tabel “farmasi\_permintaan\_unit”

- Aktivitas **Permintaan Barang**

Tabel transaksi yang digunakan *farmasi\_permintaan\_barang* kemudian untuk mendapatkan informasi ID Permintaan Barang tersimpan sama dengan tabel transaksi pada kolom *id\_farmasi\_permintaan\_unit*. Atribut *timestamp* tersimpan pada kolom *waktu\_permintaan* pada tabel transaksi. Atribut *originator* tersimpan dalam kolom *id\_petugas\_peminta* pada tabel transaksi, nama petugas tersimpan pada tabel *pengguna* dihubungkan dengan kolom *id\_pengguna*

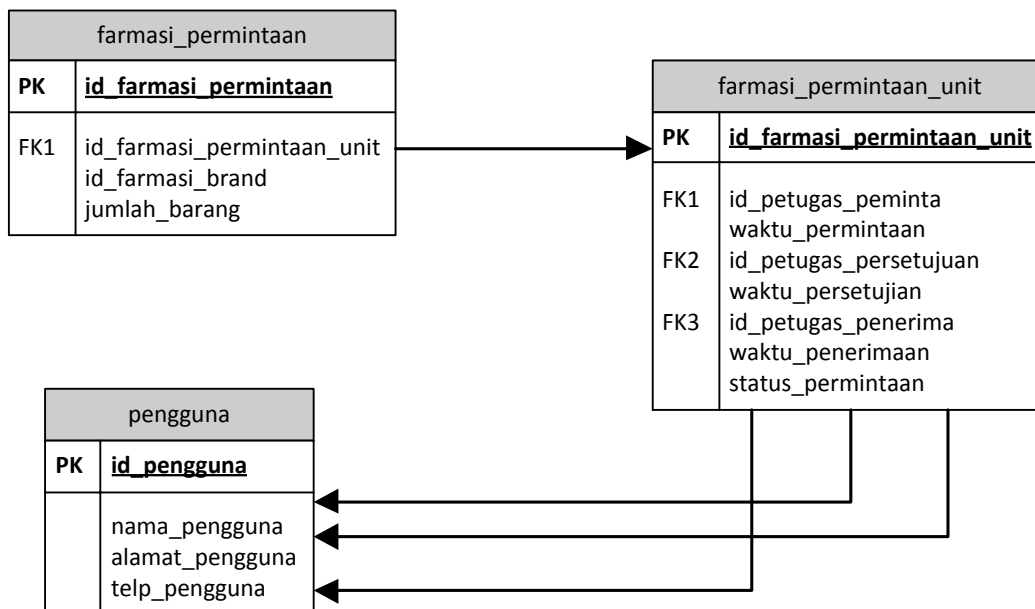
- Aktivitas **Persetujuan Barang**

Tabel transaksi yang digunakan *farmasi\_permintaan\_barang* kemudian untuk mendapatkan informasi ID Permintaan Barang tersimpan sama dengan tabel transaksi pada kolom *id\_farmasi\_permintaan\_unit*. Atribut *timestamp* tersimpan pada kolom *waktu\_persetujuan* pada tabel transaksi. Atribut *originator*

tersimpan dalam kolom *id\_petugas\_persetujuan* pada tabel transaksi, nama petugas tersimpan pada tabel *pengguna* dihubungkan dengan kolom *id\_pengguna*.

○ Aktivitas **Penerimaan Barang**

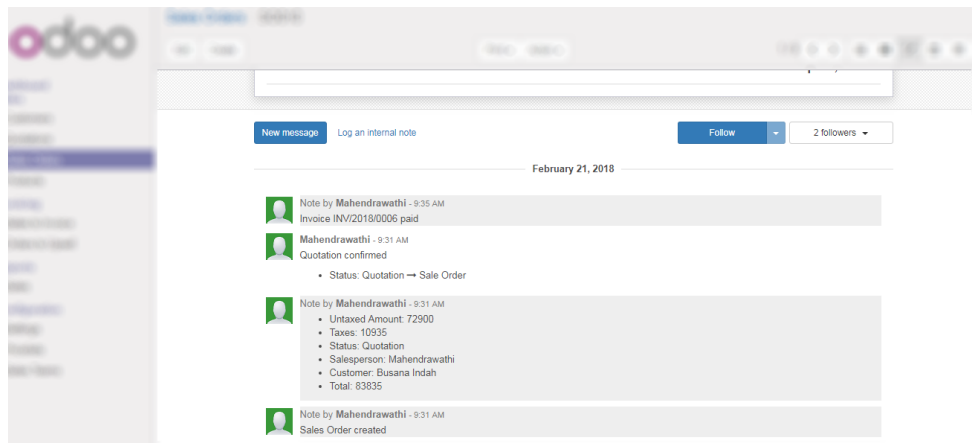
Tabel transaksi yang digunakan *farmasi\_permintaan\_barang* kemudian untuk mendapatkan informasi ID Permintaan Barang tersimpan sama dengan tabel transaksi pada kolom *id\_farmasi\_permintaan\_unit*. Atribut *timestamp* tersimpan pada kolom *waktu\_penerimaan* pada tabel transaksi. Atribut *originator* tersimpan dalam kolom *id\_petugas\_penerima* pada tabel transaksi, nama petugas tersimpan pada tabel *pengguna* dihubungkan dengan kolom *id\_pengguna*.



**Gambar 4.2** Bentuk relasi tabel atribut *event logs* pada aktivitas Permintaan barang, persetujuan permintaan dan penerimaan barang

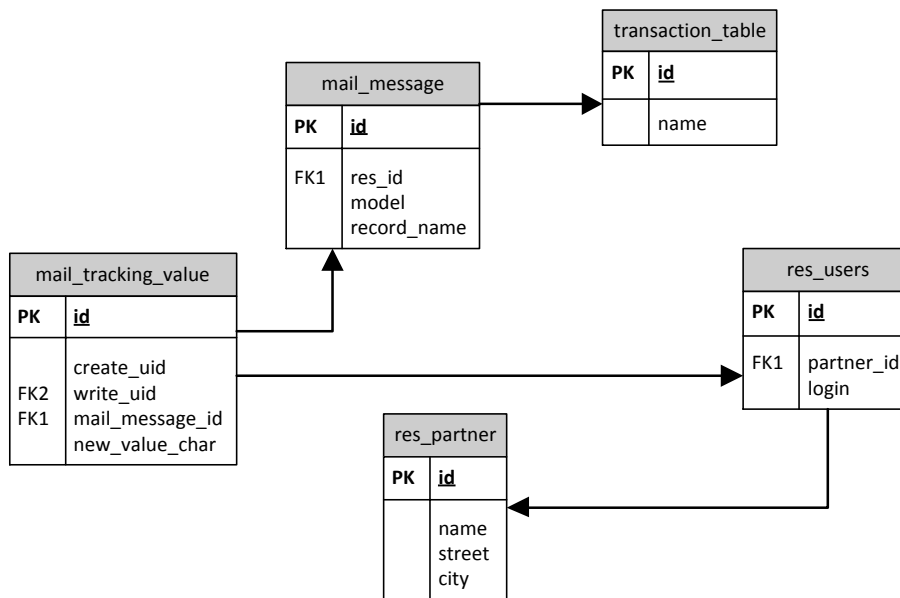
2. Atribut *event log* pada basis data Odoo ERP

Untuk dapat mengambil informasi *event log* pada Odoo ERP dapat memanfaatkan *message log* yang tersedia saat sejak awal instalasi Odoo ERP.



**Gambar 4.3** Bentuk *message log* pada sistem Odoo ERP

Cara kerja *message log* ini mencatat setiap aktivitas perubahan data oleh pengguna dalam sistem. Pencatatan ini bersifat umum untuk semua *field* pada basis data untuk tiap aktivitas. Agar dapat digunakan sesuai format standart *event log* maka perlu proses lagi agar bentuk informasi yang diambil sesuai dengan kebutuhan. Tabel yang digunakan untuk menyimpan informasi *message log* pada Odoo ERP adalah tabel *mail\_message* dan *mail\_tracking\_value*. Tabel *mail\_message* menyimpan informasi *header* tiap aktivitas yang meliputi satu alur proses kemudian tabel *mail\_tracking\_value* menyimpan deskripsi perubahan data disertai nama kolom serta nilai perubahan sebelum dan sesudah berdasarkan tabel *header mail\_message*. Karena pencatatan dalam *message\_log* ini bersifat umum sehingga untuk dapat melakukan ekstraksi *event log* perlu dikombinasi dengan tabel transaksi yang digunakan oleh Odoo ERP agar hasil yang didapatkan valid. Untuk lebih jelasnya, sumber informasi *event log* pada tiap aktivitas dalam Odoo ERP akan dijabarkan satu persatu dalam penjelasan dibawah ini.



**Gambar 4.4** Bentuk relasi sumber event log pada Odoo ERP

- **Proses Bisnis Order Penjualan**
  - Pada penjelasan sebelumnya tabel utama yang digunakan dalam yaitu *mail\_message* dan *mail\_tracking\_value* bentuk relasi kedua tabel tersebut adalah *one to many* dimana tabel *mail\_message* sebagai *parent* dan *mail\_tracking\_value* sebagai *child* yang dihubungkan oleh kolom *id* pada tabel *mail\_message* dan kolom *mail\_message\_id*. Kemudian untuk tabel lain yang dibutuhkan untuk mendapatkan aktor yang melakukan aktivitas yaitu tabel *res\_users* dan *res\_partner*, bentuk relasi kedua tabel tersebut adalah *one to one* dengan kata lain satu data di dalam tabel *res\_users* juga memiliki data dalam tabel *res\_partner*. Dalam sistem Odoo ERP aktor yang melakukan aktivitas dalam sistem selalu disimpan dalam tiap tabel transaksi dengan keterangan kolom *create\_uid* atau *write\_uid*. Selanjutnya adalah penjelasan hubungan antara tabel-tabel tersebut dengan tabel transaksi yang ditandai dengan beberapa parameter tambahan dan relasi dengan tabel transaksi sebagai pembeda untuk tiap aktivitas dalam *event log*.

- Aktivitas **Buat Penawaran**

Untuk aktivitas ini selain 4 sumber tabel yang digunakan pada penjelasan sebelumnya terdapat satu sumber tabel lagi untuk memastikan aktivitas tersebut memang tersimpan dalam transaksi. Untuk hubungan antara tabel *mail\_message* dengan *res\_users* yang diwaliki relasi pada kolom *write\_uid/create\_uid* pada tabel *mail\_message* dengan kolom *id* pada tabel *res\_users*. Tabel transaksi yang digunakan ada *sale\_order* yang berelasi dengan tabel *mail\_message* dengan kolom *id* dan kolom *res\_id* pada tabel *sale\_order*. Kemudian untuk mengetahui bahwa pencatatan sistem *message\_log* pada Odoo ERP adalah aktivitas “Buat Quotation” maka ditambahkan parameter “Quotation” pada kolom *new\_value\_char* tabel *mail\_tracking\_value* selain itu parameter “Sale Order” pada kolom *model* tabel *mail\_message*. Kemudian untuk informasi tiap atribut *event log* didefinisikan dari kolom dan tabel yang berbeda, *CaseID* diambil dari kolom *origin* pada tabel *sale\_order*, *Timestamp* diambil dari kolom *write\_date* pada tabel *mail\_tracking\_value*, kemudian untuk *Originator* diambil dari tabel *res\_partner*.

- Aktivitas **Konfirmasi Penawaran**

Sama seperti aktivitas “Buat Quotation” sumber tabel pada aktivitas ini hanya dibedakan berdasarkan parameter “Sale Order” pada kolom *new\_value\_char* tabel *mail\_tracking\_value*.

- Aktivitas **Buat Invoice**

Pada aktivitas ini erat kaitannya dengan dengan transaksi keuangan sehingga untuk mendapatkan informasi event logs pada aktivitas ini perlu melibatkan tabel transaksi yang berhubungan keuangan atau akuntansi. Tabel transaksi digunakan adalah tabel *account\_invoice* yang menyimpan tiap invoice yang terbuat dalam sistem dan tabel *account\_account* yang menyimpan jenis kode akun pada akuntansi yang menandakan bahwa transaksi tersebut adalah transaksi penjualan. Tabel transaksi *account\_invoice*

berelasi dengan tabel *mail\_message* pada kolom *res\_id* dan berelasi pada tabel *account\_account* pada dengan kolom *account\_id* tabel *account\_invoice*. Selanjutnya untuk menandakan bahwa perubahan data pada tabel *mail\_tracking\_value* adalah aktivitas “Buat Invoice” diperlukan tambahan parameter “Open” pada kolom *new\_value\_char*, parameter “Account Receivable” pada tabel *account\_account* kolom *name*, parameter “account.invoice” pada tabel *mail\_message* kolom *model*

- Aktivitas **Bayar Invoice**

Sumber informasi event log pada aktivitas ini hamper sama dengan aktivitas “Buat Invoice” namun dibedakan pada parameter “Done” pada kolom *new\_value\_char* pada tabel *mail\_message*.

- Aktivitas **Persiapan Pengiriman**

Pada proses persiapan pengiriman ini berkaitan dengan pengambilan stok yang akan dikirim, sehingga melibatkan tabel transaksi perubahan stok dalam Odoo ERP. Tabel transaksi untuk perubahan stok di Odoo ERP bernama *stock\_move* dan berhubungan dengan tabel *stock\_picking* yang menjabarkan mengenai pengambilan stok pada tiap transaksi yang berkaitan dengan stok barang. Isi dalam tabel *stock\_move* bercampur dengan transaksi lainnya contohnya manufaktur, pembelian dan perpindahan stok dari gudang. Untuk itu perlu diperjelas agar sesuai dengan aktivitas pengiriman barang, sebagai perubahan stok yang berkaitan dalam penjualan ditandai dalam tabel *procurement\_rule* dengan kolom *name* yang berisi nilai “WH: Stock -> Customers”. Tabel *procurement\_rule* ini berelasi dengan tabel *stock\_move* dengan kolom *procurement\_id*. Kemudian parameter lain yang digunakan adalah kolom *new\_value\_char* pada tabel *mail\_tracking\_value* dengan nilai “Draft” serta kolom *model* pada tabel *mail\_message* dengan nilai “stock.picking”.

- Aktivitas **Pengiriman Selesai**

Sumber informasi *event log* pada aktivitas ini sama dengan pada aktivitas “Persiapan Pengiriman” hanya dibedakan pada parameter *new\_value\_char* pada tabel *mail\_tracking\_value* dengan nilai “Done”.

- Aktivitas **Penjualan Selesai**

Untuk aktivitas ini sumber informasi event logs sama dengan pada proses “Buat Quotation” dibedakan pada parameter *new\_value\_char* pada tabel *mail\_tracking\_value* dengan nilai “Done”

#### 4.4 Pemrosesan bahasa alami ke dalam bahasa SQL

Dalam mengubah bahasa alami ke dalam bahasa SQL memiliki beberapa tahapan yang telah dijelaskan pada bab sebelumnya. Dari tahapan penentuan proses bisnis dan aktivitas kemudian mendefinisikan informasi *event logs* yang tersimpan pada tabel dan kolom dalam basis data, akan digabungkan dengan konsep bahasa alami terstruktur yang sudah ditentukan.

##### 4.4.1 Scanner (*Tokenizing*)

Setelah memnentukan bahasa alami yang digunakan sebagai masukan kemudian masukan tersebut, pada proses scanner ini terdiri dari beberapa tahapan. Pada tiap tahapan akan dijelaskan dan diimplementasikan dalam bentuk format dengan bahasa pemrograman PHP dengan Yii Framework versi 1.

1. Mengubah kalimat ke dalam huruf kecil menggunakan fungsi *strtolower*

```
1. $natural_language = strtolower($natural_language_origin);
```

2. Membersihkan kalimat dari tanda baca `[]-^.;:(){}&=>*~%?!`  Dengan menggunakan fungsi *preg\_replace*

```
1. $natural_language = preg_replace('/[^\a-zA-Z0-9*\']', '/', $natural_language);
```

3. Memecah kalimat menjadi daftar *token*

```
1. $natural_language_arr_month = explode(" ", $natural_language);
```

#### 4.4.2 Analisis Sintak (*Parser*)

Dalam sebuah kalimat Bahasa Indonesia terdiri dari beberapa kata yang tergabung menjadi sebuah kalimat. Pada masukan penelitian kali ini beberapa kata tersebut akan dikelompokkan berdasarkan tujuan proses ekstraksi *event log*. Kata-kata tersebut ada yang digunakan digunakan sebagai keterangan proses ekstraksi, ada pula beberapa kata yang tidak digunakan sehingga harus dibuang agar memudahkan dalam konversi bahasa alami menjadi syntax SQL.

**Tabel 4.4** Daftar kata yang diabaikan

KATA SAMBUNG	KATA KETERANGAN
Sampai	Berdasarkan
	Pada

Selain kata yang dapat diabaikan terdapat beberapa kata yang akan diproses yang berhubungan dengan proses ekstraksi *event log* kata tersebut dikelompokkan berdasarkan tujuan dan penggunaan masing-masing. Untuk membantu proses ekstraksi terdapat kata-kata yang digunakan sebagai pembantu dalam proses pengenalan atribut dalam bahasa alami.

**Tabel 4.5** Daftar kata pembantu

KATA PEMBANTU PERINTAH	KATA PEMBANTU KETERANGAN WAKTU
Tampilkan	Bulan
Urutkan	Tanggal
Saring	Tahun
Proses	
Aktivitas	

Dari kata pembantu tersebut diikuti dengan operator atau atribut yang digunakan sebagai nilai yang mempengaruhi hasil bentuk *event log* dalam proses ekstraksi.

**Tabel 4.6** Daftar atribut dan operator

OPERATOR ATRIBUT	OPERATOR TANGGAL	OPERATOR BULAN	OPERATOR TAHUN
CaseID	Angka 1-31	Januari	4 digit angka mulai 2011



Timestamp		Februari	
Originator		Maret	
		April	
		Mei	
		Juni	
		Juli	
		Agustus	
		September	
		Oktober	
		November	
		Desmber	

Syntax yang digunakan dalam pengambilan nilai atribut dan operator untuk perintah pengurutan data dan konversi tanggal adalah sebagai berikut

a. Mengambil token atribut waktu tanggal

```

2  /**
3   * PENGAMBILAN FILTER TANGGAL
4   */
5  $natural_language_arr_date = explode(" ", $natural_language);
6  $date_found = array();
7  $date_found_pos = array();
8  $index_arr_date = 0;
9  foreach ($natural_language_arr_date as $d) {
10     if ($d == 'tanggal') {
11         array_push($date_found_pos, $index_arr_date);
12     }
13     $index_arr_date++;
14 }

```

b. Mengambil token atribut waktu bulan

```

15 /**
16 * PENGAMBILAN FILTER BULAN
17 */
18 $natural_language_arr_month = explode(" ", $natural_language);
19 $month_found = array();
20 if (in_array('bulan', $natural_language_arr_month)) {
21     $month_found_pos = array_search('bulan', $natural_language_arr_month);
22     array_push($month_found, $natural_language_arr_month[$month_found_pos + 1]);
23 }

```

c. Mengambil token atribut waktu tahun

```

1.  /**
2.   * PENGAMBILAN FILTER TAHUN
3.   */
4.  $natural_language_arr_year = explode(" ", $natural_language);
5.  $year_found = array();
6.  if (in_array('tahun', $natural_language_arr_year)) {
7.      $year_found_pos = array_search('tahun', $natural_language_arr_year);
8.      array_push($year_found, $natural_language_arr_year[$year_found_pos + 1]);
9.  }

```

#### d. Mengambil token pengurutan hasil

```

1.  /**
2.   * PENGAMBILAN URUTAN KOLOM EVENT LOG
3.   */
4.  $natural_language_arr_urutkan = explode(" ", $natural_language);
5.  $urutkan_found = array();
6.  $urutkan_found_pos = array();
7.  $index_arr_urutkan = 0;
8.  $urutkan_list_arr = array('case id', 'caseid', 'case_id', 'originator');
9.  $urutkan_found_strpos = strpos($natural_language_origin, 'urutkan');
10. if ($urutkan_found_strpos !== false) {
11.     $natural_language_urutkan_ori = substr($natural_language_origin, $urutkan_found_strpos,
12.         strlen($natural_language_origin) - $urutkan_found_strpos);
13.     $natural_language_urutkan = str_replace(',', ' ', $natural_language_urutkan_ori);
14.     foreach ($urutkan_list_arr as $d) {
15.         $check_n1_urutkan = strpos($natural_language_urutkan, $d);
16.         if ($check_n1_urutkan !== false) {

```

#### e. Mengambil token penyaringan hasil

```

2.  /**
3.   * PENGAMBILAN FILTER KOLOM EVENT LOG
4.   */
5.  $kolom_found = array();
6.  $kolom_found_pos = array();
7.  $index_arr_kolom = 0;
8.  $kolom_list_arr = array('case id', 'caseid', 'case_id', 'timestamp', 'originator',);
9.  foreach ($kolom_list_arr as $d) {
10.     $check_n1_kolom = strpos($natural_language, $d);
11.     if ($check_n1_kolom !== false) {
12.         if ($d == 'case id' || $d == 'caseid') {
13.             $natural_language = str_replace($d, "case_id", $natural_language);
14.         }
15.     }
16. }

```

Khusus untuk proses bisnis dan aktivitas yang tersusun dari lebih dari satu kata maka lebih disebut sebagai frasa dalam Bahasa Indonesia yang terbentuk dari kata kerja dan kata benda. Beberapa frasa yang akan terdaftar dalam penelitian ini yaitu.

**Tabel 4.7** Daftar frasa yang diproses

FRASA PROSES BISNIS	FRASA AKTIVITAS
Distribusi Barang	Permintaan Barang
	Persetujuan Barang
	Penerimaan Barang
Order Penjualan	Buat Penawaran
	Konfirmasi Penawaran
	Buat Tagihan
	Bayar Tagihan
	Persiapan Pengiriman
	Selesai Pengiriman
	Selesai Penjualan

Syntax yang digunakan untuk mengambil frasa yang termasuk dalam proses bisnis dan aktivitas yang sudah ditentukan sebelumnya adalah sebagai berikut.

```

1.      /**
2.      * PENGAMBILAN PROSES BISNIS
3.      */
4.      $process_found = array();
5.      $event logs_process = SemantikController::getEvent logsProcess($source_db_id);
6.      if (count($event logs_process) > 0) {
7.          foreach ($event logs_process as $d) {
8.              $check_n1_process = strpos($natural_language, strtolower($d['process_name']))
9.          );
10.             if ($check_n1_process !== false) {
11.                 array_push($process_found, $d['process_name']);
12.                 $natural_language = str_replace(strtolower($d['process_name']), "", $nat
13.                 ural_language);
14.             }
15.         }
16.     }
17. }

18. /**
19.  * PENGAMBILAN AKTIVITAS
20.  */
21. $activity_found = array();
22. if (count($process_found) > 0) {
23.     foreach ($process_found as $d) {
24.         $id_event logs_process = Yii::app()->db->createCommand("select id_event
25.         logs_process from event logs_process where lower(process_name)='{$d}'")->queryScalar();
26.         $event logs_activity = SemantikController::getEvent
27.         logsActivity($source_db_id, $id_event logs_process);

```

```

22.         foreach ($event logs_activity as $d1) {
23.             array_push($activity_found, $d1['activity_name']);
24.         }
25.     }
26. } else {
27.     $event logs_activity = SemantikController::getEvent logsActivity($source_db_id);
28.     if (count($event logs_activity) > 0) {
29.         foreach ($event logs_activity as $d) {
30.             $check_nl_activity = strpos($natural_language, strtolower($d['activity_name']))
31.             ;
32.             if ($check_nl_activity !== false) {
33.                 array_push($activity_found, $d['activity_name']);
34.                 $natural_language = str_replace(strtolower($d['activity_name']), "", $natur
35.                 al_language);
36.             }
37.         }
38.     }

```

Setelah proses pengelompokan kata dari bahasa alami maka akan didapat beberapa kata yang disimpan untuk selanjutnya diproses dalam bentuk algoritma konversi ke dalam syntax SQL. Proses konversi dilakukan berdasarkan aturan produksi dan grammar yang dalam Bahasa Indonesia seperti penelitian yang dilakukan (Hartati dan Zuliarso, 2011).

#### 4.4.2.1 Analisa konversi unsur waktu dalam bahasa alami

Kalimat keterangan waktu adalah kalimat yang di dalamnya terdapat kata keterangan waktu sebagai penunjuk waktu terjadinya sebuah aktivitas, kegiatan, dan peristiwa tertentu. Kata keterangan yang digunakan dalam kalimat keterangan diantaranya ialah besok, pagi, siang, sore, malam, esok, kemarin, pada, lusa, penunjuk hari (senin, selasa, dan seterusnya) dan lain sebagainya. Pada penelitian ini unsur waktu yang digunakan adalah berformat tanggal, bulan dan tahun hal ini berdasarkan standart *event log* yang mendeskripsikan waktu kejadian sebuah aktivitas yang lengkap.

#### 4.4.3 Analisis Semantik (*Translator*)

Setelah proses penentuan sumber tabel dan kolom yang tersimpan dalam basis data untuk tiap basis data, maka akan dibentuk menjadi sebuah bahasa SQL yang digunakan sebagai pengambilan informasi event logs dalam basis data. Pada proses kali ini konversi SQL akan dibagi beberapa bagian yaitu bagian dasar ,

bagian penyaringan dan bagian pengurutan. Dalam bahasa SQL umumnya pengambilan informasi dalam basis data yang tersimpan didalam basis data menggunakan perintah “*select \* from nama\_tabel*” jika terdapat penyaringan data maka perlu ditambahkan perintah “*where kolom=nilai*” kemudian jika ingin mengurutkan berdasarkan nama kolom tertentu ditambahkan perintah “*order by nama\_kolom*” sehingga jika digabung menjadi “*select \* from nama\_tabel where nama\_kolom=nilai order by nama\_kolom*”. Perintah SQL tersebut dikonversi oleh komputer untuk menampilkan informasi yang tersimpan basis data untuk semua kolom pada nama tabel “*nama\_tabel*” dengan tambahan penyaringan data berdasarkan “*nama\_kolom*” dengan “*nilai*” tertentu dan pengurutan data berdasarkan kolom “*nama\_kolom*”. Pada penelitian ini perintah SQL tersebut dibagi menjadi 3 sesuai dengan perintah yang telah didefinisikan dalam bahasa alami. Pembagian tersebut dijelaskan sebagai berikut

1. Bagian dasar (B1)

Kata perintah bahasa alami “*Tampilkan*” mewakili bahasa SQL “*select \* from nama\_tabel*” nama kolom yang ditampilkan ada atribut *event log* yaitu *case\_id*, *aktivitas*, *timestamp* dan *originator*

2. Bagian penyaringan (B2)

Kata perintah bahasa alami “*Saring*” mewakili bahasa SQL “*where nama\_kolom=nilai*”

3. Bagian pengurutan (B3)

Kata perintah bahasa alami “*Urutkan*” mewakili bahasa SQL “*order by nama\_kolom*”

Dari ketiga bagian translasi perintah bahasa ke dalam bahasa SQL kemudian penyesuaian dengan semua KPE yang sudah ditentukan sebelumnya. Kemudian pada setiap bagian translasi disesuaikan dengan nama tabel dan kolom pada setiap proses bisnis dan aktivitas untuk tiap sistem. Sebelum proses penggabungan translasi dengan semua KPE maka perlu ditentukan terlebih dahulu untuk penentuan translasi bahasa SQL untuk setiap aktivitas yang sudah ditentukan sebelumnya. Pada penjelasan bagian 4.3.1 telah ditentukan sumber tabel dan kolom yang sesuai dengan atribut *event log* dari sumber tersebut akan dibentuk translasi berdasarkan bagian perintah yang ada. Pertama penentuan bentuk

translasi SQL untuk bagian dasar (B1) pada tiap aktivitas dalam proses bisnis tersebut akan dijelaskan pada bagian dibawah ini

- **Proses bisnis Distribusi Barang (SIM RS)**

- **Aktivitas Permintaan barang**

Bahasa SQL yang terbentuk :

```
“select farmasi_permintaan_unit.id_farmasi_permintaan_unit as  
caseid, ‘Permintaan Barang’ as activity,  
farmasi_permintaan_unit.waktu_permintaan as timestmp,  
pengguna.nm_pengguna as originator  
From farmasi_Permintaan_unit  
Join pengguna on  
pengguna.id_pengguna=farmasi_permintaan_unit.id_petugas_peminta  
”
```

- **Aktivitas Persetujuan barang**

Bahasa SQL yang terbentuk :

```
“select farmasi_permintaan_unit.id_farmasi_permintaan_unit as  
caseid, ‘Permintaan Barang’ as activity,  
farmasi_permintaan_unit.waktu_persetujuan as timestmp,  
pengguna.nm_pengguna as originator  
From farmasi_Permintaan_unit  
Join pengguna on  
pengguna.id_pengguna=farmasi_permintaan_unit.id_petugas_persetu  
juan”
```

- **Aktivitas Penerimaan Barang**

Bahasa SQL yang terbentuk :

```
“select farmasi_permintaan_unit.id_farmasi_permintaan_unit as  
caseid, ‘Permintaan Barang’ as activity,  
farmasi_permintaan_unit.waktu_penerimaan as timestmp,  
pengguna.nm_pengguna as originator  
From farmasi_Permintaan_unit  
Join pengguna on  
pengguna.id_pengguna=farmasi_permintaan_unit.id_petugas_penerima  
an”
```

- **Proses bisnis Order Penjualan (Odoo ERP)**

- **Aktivitas Buat Penawaran**

Bahasa SQL yang terbentuk :

```
“SELECT sale_order. NAME AS case_id, 'Buat Quotation' AS
activity, to_char(mail_tracking_value.write_date, 'YYYY-MM-DD
HH24:MI:SS') AS timestmp, res_partner. NAME AS originator
FROM mail_tracking_value
JOIN mail_message ON TRIM (mail_message. ID :: TEXT) = TRIM (
mail_tracking_value.mail_message_id :: TEXT)
JOIN res_users ON TRIM (res_users. ID :: TEXT) = TRIM
(mail_message.write_uid :: TEXT)
JOIN res_partner ON TRIM (res_partner. ID :: TEXT) = TRIM
(res_users.partner_id :: TEXT)
JOIN sale_order ON TRIM (sale_order. ID :: TEXT) = TRIM
(mail_message.res_id :: TEXT)”
```

- **Aktivitas Konfirmasi Penawaran**

Bahasa SQL yang terbentuk :

```
“SELECT sale_order. NAME AS case_id, 'Konfirmasi Quotation' AS
activity, to_char( mail_tracking_value.write_date, 'YYYY-MM-DD
HH24:MI:SS' ) AS timestmp, res_partner. NAME AS originator
FROM mail_tracking_value
JOIN mail_message ON TRIM (mail_message. ID :: TEXT) = TRIM (
mail_tracking_value.mail_message_id :: TEXT )
JOIN res_users ON TRIM (res_users. ID :: TEXT) = TRIM (
mail_message.write_uid :: TEXT )
JOIN res_partner ON TRIM (res_partner. ID :: TEXT) = TRIM
(res_users.partner_id :: TEXT)
JOIN sale_order ON TRIM (sale_order. ID :: TEXT) = TRIM
(mail_message.res_id :: TEXT)”
```

- **Aktivitas Buat Tagihan**

Bahasa SQL yang terbentuk :

```
“SELECT account_invoice.origin AS case_id, 'Buat Invoice' AS
activity, to_char( mail_tracking_value.write_date, 'YYYY-MM-DD
HH24:MI:SS' ) AS timestmp, res_partner. NAME AS originator
FROM mail_tracking_value
```

```

JOIN mail_message ON TRIM (mail_message. ID :: TEXT) = TRIM (
mail_tracking_value.mail_message_id :: TEXT)
JOIN account_invoice ON TRIM (account_invoice. ID :: TEXT) =
TRIM (mail_message.res_id :: TEXT)
JOIN account_account ON TRIM (account_account. ID :: TEXT) =
TRIM ( account_invoice.account_id :: TEXT )
JOIN res_users ON TRIM (res_users. ID :: TEXT) = TRIM (
mail_message.write_uid :: TEXT )
JOIN res_partner ON TRIM (res_partner. ID :: TEXT) = TRIM
(res_users.partner_id :: TEXT)”

```

#### ○ **Aktivitas Bayar Tagihan**

Bahasa SQL yang terbentuk :

```

“SELECT account_invoice.origin AS case_id, 'Buat Invoice' AS
activity, to_char( mail_tracking_value.write_date, 'YYYY-MM-DD
HH24:MI:SS' ) AS timestmp, res_partner. NAME AS originator
FROM mail_tracking_value
JOIN mail_message ON TRIM (mail_message. ID :: TEXT) = TRIM (
mail_tracking_value.mail_message_id :: TEXT)
JOIN account_invoice ON TRIM (account_invoice. ID :: TEXT) =
TRIM (mail_message.res_id :: TEXT)
JOIN account_account ON TRIM (account_account. ID :: TEXT) =
TRIM ( account_invoice.account_id :: TEXT )
JOIN res_users ON TRIM (res_users. ID :: TEXT) = TRIM (
mail_message.write_uid :: TEXT )
JOIN res_partner ON TRIM (res_partner. ID :: TEXT) = TRIM
(res_users.partner_id :: TEXT)”

```

#### ○ **Aktivitas Persiapan Pengiriman**

Bahasa SQL yang terbentuk :

```

“SELECT stock_move.origin AS case_id, 'Persiapan Pengiriman'
AS activity, to_char( stock_move.write_date, 'YYYY-MM-DD
HH24:MI:SS' ) AS timestmp, res_partner. NAME AS originator
FROM mail_tracking_value
JOIN mail_message ON TRIM (mail_message. ID :: TEXT) = TRIM (
mail_tracking_value.mail_message_id :: TEXT )
JOIN stock_picking ON TRIM (stock_picking. ID :: TEXT) = TRIM
(mail_message.res_id :: TEXT)

```



```

JOIN stock_move ON TRIM ( stock_move.picking_id :: TEXT ) =
TRIM (stock_picking. ID :: TEXT)
JOIN res_users ON TRIM (res_users. ID :: TEXT) = TRIM (
mail_message.write_uid :: TEXT )
JOIN res_partner ON TRIM (res_partner. ID :: TEXT) = TRIM
(res_users.partner_id :: TEXT)
JOIN procurement_rule ON TRIM (procurement_rule. ID :: TEXT) =
TRIM (stock_move.rule_id :: TEXT)”

```

#### ○ **Aktivitas Selesai Pengiriman**

Bahasa SQL yang terbentuk :

```

“SELECT stock_move.origin AS case_id, 'Persiapan Pengiriman'
AS activity, to_char( stock_move.write_date, 'YYYY-MM-DD
HH24:MI:SS' ) AS timestmp, res_partner. NAME AS originator
FROM mail_tracking_value
JOIN mail_message ON TRIM (mail_message. ID :: TEXT) = TRIM (
mail_tracking_value.mail_message_id :: TEXT )
JOIN stock_picking ON TRIM (stock_picking. ID :: TEXT) = TRIM
(mail_message.res_id :: TEXT)
JOIN stock_move ON TRIM ( stock_move.picking_id :: TEXT ) =
TRIM (stock_picking. ID :: TEXT)
JOIN res_users ON TRIM (res_users. ID :: TEXT) = TRIM (
mail_message.write_uid :: TEXT )
JOIN res_partner ON TRIM (res_partner. ID :: TEXT) = TRIM
(res_users.partner_id :: TEXT)
JOIN procurement_rule ON TRIM (procurement_rule. ID :: TEXT) =
TRIM (stock_move.rule_id :: TEXT)”

```

#### ○ **Aktivitas Selesai Penjualan**

Bahasa SQL yang terbentuk :

```

“SELECT sale_order. NAME AS case_id, 'Selesai Penjualan' AS
activity, to_char( mail_tracking_value.write_date, 'YYYY-MM-DD
HH24:MI:SS' ) AS timestmp, res_partner. NAME AS originator
FROM mail_tracking_value
JOIN mail_message ON TRIM (mail_message. ID :: TEXT) = TRIM (
mail_tracking_value.mail_message_id :: TEXT )
JOIN res_users ON TRIM (res_users. ID :: TEXT) = TRIM (
mail_message.write_uid :: TEXT )

```

```
JOIN res_partner ON TRIM (res_partner. ID :: TEXT) = TRIM
(res_users.partner_id :: TEXT)
JOIN sale_order ON TRIM (sale_order. ID :: TEXT) = TRIM
(mail_message.res_id :: TEXT)”
```

Setelah pendefinisian translasi bahasa SQL untuk bagian dasar selanjutnya adalah bagian penyaringan yang disesuaikan dengan bahasa alami. Bagian penyaringan ini bahasa SQL yang digunakan diawali dengan “*where*” yang kemudian diikuti dengan nama kolom beserta nilai sebagai parameter dalam penyaringan data. Proses translasi bagian penyaringan ini dibagi menjadi 2 yaitu penyaringan yang berasal dari perintah bahasa alami, dengan penyaringan yang berasal dari pendefinisian sumber tabel dan kolom pada beberapa aktivitas dalam proses bisnis. Contohnya pada aktivitas yang terdapat pada sistem Odoo ERP untuk mendapatkan informasi *event log* yang sesuai perlu ada tambahan beberapa parameter khusus yang nantinya akan translasi ke dalam bahasa SQL. Bentuk translasi pada bagian penyaringan tersebut yaitu :

#### 1. Penyaringan dari Bahasa Alami

- Untuk unsur waktu

Seperti yang dijelaskan pada **tabel 4.5** bahwa bahasa alami yang digunakan untuk kata keterangan waktu adalah tanggal, bulan dan tahun. Penyaringan ini berdasarkan pada atribut timestamp pada *event log* karena nilai pada atribut adalah waktu kejadian yang memiliki didefinisikan dalam bentuk format waktu. Tiap atribut tersebut memiliki nilai yang di jabarkan pada **tabel 4.6**, kata keterangan waktu dalam bahasa alami ini dimasukan pada bagian penyaringan agar dapat membentuk bahasa SQL yang dimengerti oleh komputer.

- Untuk kata keterangan tanggal

Menyatakan sebagai sebuah periode yang diawali tanggal awal dan tanggal akhir. Format tanggal yang digunakan dalam bahasa alami adalah (D<spasi>M<spasi>Y) dimana D=operator tanggal, dan M operator bulan dan Y = operator tahun, nilai tiap operator

tersebut dibatasi sesuai dengan **tabel 4.6**. bahasa SQL yang terbentuk sebagai penyaringan pada *event log* adalah

*“where timestmp between operator\_tanggal\_awal\_periode and operator\_tanggal\_akhir\_periode”*

- Untuk kata keterangan bulan

Menyatakan sebuah periode yang dibatasi dengan bulan tertentu yang didefinisikan pada **tabel 4.6** pada operator bulan. Pada penelitian ini batasan bahasa SQL untuk DBMS PostgreSQL sehingga translasi ke dalam bahasa SQL fungsi yang digunakan yaitu *“month(nama\_kolom)=format\_bulan”*. Untuk format bulan yang dapat diproses fungsi tersebut adalah 01-12 sehingga dari operator bulan pada **tabel 4.6** perlu ada fungsi konversi tambahan. Contoh konversi tersebut diantaranya untuk operator bulan januari dikonversi menjadi 01, untuk operator bulan februari dikonversi menjadi 02, untuk operator bulan maret dikonversi menjadi 03 dan seterusnya sampai dengan desember menjadi 12. Sehingga dapat membentuk translasi bahasa SQL bagian penyaringan sebagai berikut

*“where month(timestmp)='01'”* untuk operator bulan januari

*“where month(timestmp)='02'”* untuk operator bulan februari

*“where month(timestmp)='03'”* untuk operator bulan maret

.....

*“where month(timestmp)='12'”* untuk operator bulan desember

- Untuk kata keterangan tahun

Menyatakan sebuah periode yang dibatasi dengan bulan tertentu yang didefinisikan pada **tabel 4.6** pada operator tahun. Format tahun yang digunakan yaitu terdiri dari 4 digit angka mulai dari tahun 1900 sampai dengan tahun sekarang. Fungsi bahasa SQL pada DBMS PostgreSQL yang dapat digunakan untuk penyaringan operator tahun yaitu *“year(nama\_kolom)=format\_tahun”*. Pada proses translasi ini tidak perlu adanya konversi tambahan, karena format tahun yang

dibutuhkan pada fungsi tersebut sama dengan pendefinisian operator tahun pada bahasa alami sehingga akan terbentuk bahasa SQL untuk bagian penyaringan sebagai berikut

“**where** year(*timestamp*)=**'1900'**” untuk operator tahun 1900

.....

“**where** month(*timestamp*)=**'2017'**” untuk operator tahun 2017

- Untuk atribut event log

Sesuai yang didefinisikan pada **tabel 4.5** bahasa alami yang digunakan dalam bagian penyaringan ini adalah “*Saring*” kemudian diikuti dengan nama atribut *event log* dengan nilai yang ditentukan. Pada atribut *activity* proses translasi ke dalam bahasa SQL sudah ditentukan pada bagian dasar sama halnya dengan atribut *timestamp* yang dijabarkan pada bagian penyaringan untuk unsur waktu. Sehingga pada penyaringan berdasarkan atribut *event log* ini hanya terbatas pada atribut *caseID* dan *Originator*. Nilai dari atribut tersebut didefinisikan berdasarkan **tabel 4.3** , sehingga dari penjabaran tersebut akan terbentuk bahasa SQL sebagai berikut :

“**where** caseid=**'nilai\_atribut'**” untuk atribut *caseID*

“**where** originator=**'nilai\_atribut'**” untuk atribut *originator*

## 2. Penyaringan dari Sumber tabel dan kolom

Proses penyaringan ini berdasarkan pendefinisian sumber tabel dan sumber kolom pada tiap aktivitas dalam proses bisnis. Tidak semua aktivitas perlu adanya proses penyaringan contohnya pada SIM RS untuk semua aktivitas dalam proses bisnis Distribusi Barang dan Pemberian E-Resep, namun pada Odoo ERP perlu ada tambahan penyaringan agar informasi event log yang didapatkan sesuai. Pada Odoo ERP tiap aktivitas dan proses bisnis memiliki parameter yang berbeda-beda sebagai penyaringan.

- **Proses bisnis Order Penjualan (Odoo ERP)**

Sesuai pada penjelasan bagian 4.3.1 untuk order penjualan tambahan parameter yang digunakan sebagai penyaringan yaitu tabel *mail\_tracking\_value* kolom *new\_value\_char* serta tabel *mail\_message*

pada kolom *model* selanjutnya untuk tiap aktivitas dibedakan yang dijabarkan sebagai berikut :

- **Aktivitas Buat Penawaran**

Penyaringan yang digunakan yaitu pada tabel *auditlog\_log\_line* kolom *state* dengan nilai 'draft' sehingga translasi ke dalam bahasa SQL yang terbentuk :

```
"WHERE mail_tracking_value.write_date IS NOT NULL  
AND mail_message.model = 'sale.order'  
AND mail_tracking_value.new_value_char = 'Quotation'"
```

- **Aktivitas Konfirmasi Penawaran**

Penyaringan yang digunakan yaitu pada tabel *auditlog\_log\_line* kolom *state* dengan nilai 'purchase' sehingga translasi ke dalam bahasa SQL yang terbentuk :

```
"WHERE mail_tracking_value.write_date IS NOT NULL  
AND mail_message.model = 'sale.order'  
AND mail_tracking_value.new_value_char = 'Sale Order'"
```

- **Aktivitas Buat Tagihan**

Penyaringan yang digunakan yaitu pada tabel *auditlog\_log\_line* kolom *state* dengan nilai 'done' sehingga translasi ke dalam bahasa SQL yang terbentuk :

```
"WHERE mail_tracking_value.write_date IS NOT NULL  
AND account_account.NAME = 'Account Receivable'  
AND mail_message.model = 'account.invoice'  
AND mail_tracking_value.new_value_char = 'Open'"
```

- **Aktivitas Bayar Tagihan**

Penyaringan yang digunakan yaitu pada tabel *auditlog\_log\_line* kolom *state* dengan nilai 'done' sehingga translasi ke dalam bahasa SQL yang terbentuk :

```
"WHERE mail_tracking_value.write_date IS NOT NULL  
AND account_account.NAME = 'Account Receivable'  
AND mail_message.model = 'account.invoice'  
AND mail_tracking_value.new_value_char = 'Paid'"
```

- **Aktivitas Persiapan Pengiriman**

Penyaringan yang digunakan yaitu pada tabel *auditlog\_log\_line* kolom *state* dengan nilai 'done' sehingga translasi ke dalam bahasa SQL yang terbentuk :

```
"WHERE stock_move.write_date IS NOT NULL
AND mail_message.model = 'stock.picking'
AND mail_tracking_value.new_value_char = 'Draft'
AND procurement_rule. NAME = 'WH: Stock -> Customers'"
```

- **Aktivitas Selesai Pengiriman**

Penyaringan yang digunakan yaitu pada tabel *auditlog\_log\_line* kolom *state* dengan nilai 'done' sehingga translasi ke dalam bahasa SQL yang terbentuk :

```
"WHERE stock_move.write_date IS NOT NULL
AND mail_message.model = 'stock.picking'
AND mail_tracking_value.new_value_char = 'Done'
AND procurement_rule. NAME = 'WH: Stock -> Customers'"
```

- **Aktivitas Penjualan Selesai**

Penyaringan yang digunakan yaitu pada tabel *auditlog\_log\_line* kolom *state* dengan nilai 'done' sehingga translasi ke dalam bahasa SQL yang terbentuk :

```
"WHERE mail_tracking_value.write_date IS NOT NULL
AND mail_message.model = 'sale.order'
AND mail_tracking_value.new_value_char = 'Done'"
```

Proses terakhir translasi adalah pada bagian pengurutan yang mewakili kalimat perintah 'Urutkan'. Proses pengurutan ini digunakan untuk membantu para analis bisnis dalam melihat hasil ekstraksi *event log* yang tersimpan dalam sebuah basis data sebelum digunakan untuk kebutuhan proses mining. Dalam standart *event log* bentuk informasi yang ditampilkanurut berdasarkan *caseID* yang merupakan identifikasi kasus dalam satu proses bisnis kemudian diurutkan sesuai waktu kejadian tiap aktivitas masing-masing. Oleh karena itu pada penelitian ini secara *default* tampilan hasil ekstraksi akan diurutkan berdasarkan atribut *caseID* dan *timestamp* namun untuk memudahkan analis bisnis, pengurutan ini dapat

disesuaikan untuk semua atribut *event log* yang ada. Dalam bahasa SQL pengurutan pada kolom tertentu bisa diatur berdasarkan urutan ‘awal’ ke ‘akhir’ yang diwakili dengan fungsi pengurutan ‘asc’ dan ‘urutan’ akhir ke ‘awal’ yang diwakili dengan fungsi pengurutan ‘desc’. Proses pengurutan tersebut otomatis disesuaikan dengan nilai kolom masing-masing yang terbagi menjadi ‘character’, ‘timestamp/data’ dan ‘numeric’. Selanjutnya translasi bahasa SQL yang terbentuk pada bagian penyaringan ini sebagai adalah:

- “**order by caseid asc**” mewakili atribut event logs *caseID* dengan metode pengurutan ‘awal’-‘akhir’
- “**order by caseid desc**” mewakili atribut event logs *caseID* dengan metode pengurutan ‘akhir’-‘awal’
- “**order by activity asc**” mewakili atribut event logs *activity* dengan metode pengurutan ‘awal’-‘akhir’
- “**order by activity desc**” mewakili atribut event logs *activity* dengan metode pengurutan ‘akhir’-‘awal’
- “**order by timestmp asc**” mewakili atribut event logs *timestamp* dengan metode pengurutan ‘awal’-‘akhir’
- “**order by timestmp desc**” mewakili atribut event logs *timestamp* dengan metode pengurutan ‘akhir’-‘awal’
- “**order by originator asc**” mewakili atribut event logs *originator* dengan metode pengurutan ‘awal’-‘akhir’
- “**order by originator desc**” mewakili atribut event logs *originator* dengan metode pengurutan ‘akhir’-‘awal’

Setelah bagian dasar, bagian penyaringan dan bagian pengurutan telah selesai di translasi. Kemudian ketiga bagian tersebut digabung menjadi bahasa SQL yang utuh yang nantinya digunakan untuk mengambil informasi *event logs*. Agar informasi event logs dapat tergabung menjadi informasi yang utuh tiap proses bisnis dibutuhkan fungsi bahasa SQL yang disebut ‘union’. Fungsi tersebut untuk menggabungkan sintak *query* satu dengan sintak *query* lain. Contohnya dalam proses bisnis permintaan barang perlu penggabungan antara sintak untuk aktivitas Permintaan Barang, Persetujuan Barang dan Penerimaan Barang. Setelah

penggabungan dengan fungsi ‘*union*’ kemudian kumpulan sintak tersebut akan dikemas menggunakan metode ‘*subquery*’ dalam bahasa SQL. Metode tersebut digunakan agar hasil ekstraksi dalam beberapa aktivitas dapat diproses kembali sesuai dengan bagian penyaringan dan bagian pengurutan. Contoh pada proses bisnis Distribusi Barang pada SIM RS hasil akhir translasi bahasa SQL tersebut menjadi

```

“SELECT * FROM (SELECT * FROM ( SELECT
farmasi_permintaan_unit.id_farmasi_permintaan_unit as
case_id, 'Permintaan Barang' as
activity, farmasi_permintaan_unit.waktu_permintaan as
timestamp, pengguna.nm_pengguna as originator FROM
farmasi_permintaan_unit JOIN pengguna on
pengguna.id_pengguna=farmasi_permintaan_unit.id_petugas_peminta)
AS A1 UNION ALL ( SELECT
farmasi_permintaan_unit.id_farmasi_permintaan_unit as
case_id, 'Persetujuan Permintaan' as
activity, farmasi_permintaan_unit.waktu_persetujuan as
timestamp, pengguna.nm_pengguna as originator FROM
farmasi_permintaan_unit JOIN pengguna on
pengguna.id_pengguna=farmasi_permintaan_unit.id_petugas_persetujuan)
UNION ALL ( SELECT
farmasi_permintaan_unit.id_farmasi_permintaan_unit as
case_id, 'Penerimaan Barang' as
activity, farmasi_permintaan_unit.waktu_penerimaan as
timestamp, pengguna.nm_pengguna as originator FROM
farmasi_permintaan_unit JOIN pengguna on
pengguna.id_pengguna=farmasi_permintaan_unit.id_petugas_penerima)
) AS B”

```

Dari hasil akhir penggabungan translasi bahasa SQL tersebut akan dibentuk berdasarkan jenis KPE yang sudah ditentukan pada penelitian ini. Kemudian pada setiap jenis KPE akan diuji untuk semua proses bisnis dan aktivitas yang ada. Proses pengujian hasil translasi bahasa SQL ini akan dibahas pada tahapan analisis pragmatik.



#### 4.4.4 Analisis Pragmatik (*Evaluator*)

Pada tahapan terakhir ini akan memeriksa hasil translasi apakah sudah sesuai dengan kaidah yang dibenarkan. Pada penelitian ini akan memeriksa token yang diambil untuk **kata pembantu keterangan waktu** (tanggal, bulan dan tahun) dengan **operator waktu**(tanggal bulan tahun), **frasa aktivitas dan proses bisnis**, **perintah penyaringan** ('Saring') untuk **operator atribut** event log, serta **perintah pengurutan** ('Urutkan') pada untuk **operator atribut** event log. Contohnya pada **kata pembantu keterangan waktu** bulan **operator waktu** yang digunakan adalah (Januari, Februari...,Desember) . Kemudian untuk **kata pembantu keterangan waktu** tanggal **operator waktu** yang digunakan dengan format '30 Januari 2017'. Potongan syntax yang digunakan dalam penelitian ini terangkum pada penjelasan bagian 4.4.1 dalam proses scanner. Pada tahap tersebut melakukan proses pencarian token yang sesuai disertai pengecekan nilai operator yang telah ditentukan dalam penelitian ini. Jika terdapat kesalahan dalam penulisan token atau bahasa alami maka sistem ini akan memberikan keterangan error.

#### 4.5 Perancangan alat ekstraksi dengan bahasa alami

Sesuai dengan rancangan yang telah dijelaskan pada bagian 3.7 pada desain sistem dan alur proses ekstraksi yang memiliki 3 macam aktor yaitu Admin Sistem, Admin *Evenl* dan Analis Bisnis kemudian pada tahapan ini akan dibahas lebih lanjut dalam pembuatan desain basis data aplikasi *Evenl* dan antarmuka yang akan dijadikan sebagai rancangan untuk membangun aplikasi *Evenl*

##### 4.5.1 Perancangan Basis data

Perancangan basis data ini bertujuan untuk membantu penyimpanan informasi yang digunakan dalam proses ekstraksi *event log*. Data yang disimpan dalam basis data tidak menyimpan data sistem yang dituju dalam arti informasi yang ada dalam basis data ini hanya digunakan sebagai pembantu proses ekstraksi *event log*. Pada proses perancangan basis data terdiri dari 3 bagian yaitu identifikasi entitas, identifikasi relasi dan identifikasi atribut dan entitas.

#### 4.5.1.1 Identifikasi Entitas

Entitas yang didefinisikan dalam aplikasi *Evenl* didasarkan pada alur proses dan desain rancangan sistem. Sesuai dengan aktivitas tiap aktor dalam aplikasi *Evenl* beberapa entitas yang didapatkan diantaranya :

**Tabel 4.8** Identifikasi Entitas aplikasi *Evenl*

<b>Nama Entitas</b>	<b>Deskripsi</b>	<b>Alias</b>	<b>Kejadian</b>
User	Merupakan Entitas yang digunakan untuk menyimpan informasi pengguna aplikasi	<i>users</i>	Setiap pengguna untuk dapat menggunakan aplikasi harus melakukan login ke dalam aplikasi
Role	Merupakan entitas yang mendefinisikan jenis aktor dalam aplikasi	<i>roles</i>	Setelah berhasil melakukan login aplikasi akan membagi fungsi aplikasi berdasarkan aktivitas sesuai jenis aktor
Klien	Merupakan entitas yang digunakan untuk menyimpan informasi klien analis bisnis maupun pemilik sistem	<i>Event logs clien</i>	Setiap partner memiliki hak akses ke dalam sistem yang dituju berdasarkan permintaan ekstraksi
Proses Bisnis	Merupakan entitas yang digunakan untuk mendefinisikan proses bisnis didalam aplikasi	<i>event logs process</i>	Setiap proses bisnis yang dipilih akan menentukan hasil informasi <i>event log</i>
Aktivitas	Merupakan entitas yang digunakan untuk mendefinisikan aktivitas berdasarkan proses bisnis	<i>event logs activity</i>	Setiap aktivitas yang dipilih akan menentukan hasil informasi <i>event logs</i>
Atribut	Merupakan entitas	<i>event logs atribut</i>	Setiap atribut digunakan

<i>event log</i>	yang digunakan untuk menyimpan informasi atribut <i>event log</i>		sebagai dasar proses ekstraksi yang diambil dari basis data sistem
Sumber tabel <i>event log</i>	Merupakan entitas yang digunakan untuk menyimpan informasi sumber tabel yang tersimpan dalam basis data sistem	<i>event logs source tables</i>	Setiap tabel yang digunakan akan menentukan bentuk bahasa SQL yang dipakai untuk proses ekstraksi
Sumber kolom <i>event log</i>	Merupakan entitas yang digunakan pendefinisikan sumber kolom pada tabel untuk setiap atribut <i>event log</i>	<i>Event logs source columns</i>	Setiap kolom yang didefinisikan akan direlasikan dengan atribut <i>event log</i>
Sumber Parameter <i>event log</i>	Merupakan entitas yang digunakan untuk menyimpan informasi parameter tambahan pada sumber tabel dan kolom untuk proses ekstraksi	<i>event logs source paramters</i>	Setiap parameter yang didefinisikan akan menentukan hasil bahasa SQL yang terbentuk
Sumber Basis Data	Merupakan entitas yang digunakan untuk menyimpan informasi akses ke basis data sistem	<i>source database</i>	Setiap sumber basis data akan dijadikan untuk melakukan koneksi ke dalam basis data sistem
Jenis Vendor DBMS	Menampilkan informasi jenis vendor untuk basis data		Setiap proses ekstraksi jenis vendor DBMS akan menentukan jenis bahasa SQL yang digunakan
Sumber Tabel Basis data	Merupakan entitas yang digunakan untuk menyimpan informasi	<i>source database_table</i>	Setiap proses penentuan sumber tabel dan kolom ekstraksi <i>event log</i> yang

	struktur tabel pada basis data sistem		menghasilkan bahasa SQL
Sumber Kolom Baiss Data	Merupakan entitas yang digunakan untuk menyimpan informasi struktur kolom pada table untuk basis data sistem	<i>source database column</i>	Setiap proses penentuan sumber tabel dan kolom ekstraksi event log yang menghasilkan bahasa SQL

#### 4.5.1.2 Identifikasi Relasi

Dari beberapa entitas tersebut akan ditentukan hubungan antar entitas sehingga membentuk sebuah basis data yang utuh dan menghindari duplikasi data. Pada proses ini untuk menggambarkan jenis relasi antara entitas yang ada, selain itu juga digunakan untuk menentukan *multiplicity constraints* (MC) dari tipe relasi. Bentuk relasi yang terbentuk pada tiap entitas tersebut adalah

**Tabel 4.9** Relasi antar entitas aplikasi *Eventl*

Nama Entitas	MC	Relasi	Nama Entitas	MC
Users	1.1	Punya	Role	1.1
Event logs client	1.1	Punya	User	1.1
	1.*	Memiliki	Event logs process	*.1
	1.*	Memiliki	Source database	*.1
Event logs process	1.*	Terdiri dari	Event logs activity	
Event logs activity	1.*	Memiliki	Event logs source tables	*.1
	1.*	Memiliki	Event logs source colums	*.1
	1.*	Memiliki	Event logs source parameter	
Event logs attribute	1.*	Memiliki	Event logs source colums	*.1
Source database	1.1	Menjabarkan	Source database vendor	
	1,*	Memiliki	Source database table	
	1.*	Memiliki	Source database colums	
Source database table	1.*	Memiliki	Source database columns	*.1
	1.*	Dimuat	Event logs source tables	

Source database column	1.*		Event logs source column	
	1.*	Dimuat	Event logs source paramters	

#### 4.5.1.3 Identifikasi Atribut dan Entias

Dari setiap entitas dan relasi antar entitas akan dibuat atribut untuk setiap entitas yang ada. Penentuan atribut ini berdasarkan pada kebutuhan akan proses ekstraksi *event log*.

**Tabel 4.10** Atribut untuk tabel users

Atribut	Deskripsi	Panjang dan tipe data	Null	Jenis Key
Id_user	Kode pengguna	Integer (11)	NO	Primary
Id_role	Kode Role	Integer (11)	NO	Foreign
Username	Username pengguna	Varchar (100)	NO	
Password	Autentifikasi password pengguna	Varchar (250)	NO	
Email	Email pengguna	Varchar (200)	YES	
Tgl_lahir	Tanggal lahir	Date	YES	
Nama_user	Nama Pengguna	Varchar (100)	NO	

**Tabel 4.11** Atribut untuk tabel roles

Atribut	Deskripsi	Panjang dan tipe data	Null	Jenis Key
Id_roles	Kode rolle	Integer (11)	NO	Primary
Role_name	Nama jenis role	Vachar (100)	NO	

**Tabel 4.12** Atribut untuk tabel event logs client

Atribut	Deskripsi	Panjang dan tipe data	Null	Jenis Key
Id_event logs_client	Kode Klien	Integer (11)	NO	Primary
Id_user	Kode User	Integer (11)	NO	Foreign
Client_name	Nama Klien	Varchar (100)	NO	
Client_from	Asal instansi /	Varchar (255)	YES	

	Perusahaan			
Client_addr	Alamat klien	Varchar (255)	YES	
Client_phone	Nomer kontak klien			

**Tabel 4.13** Atribut untuk tabel event logs process

Atribut	Deskripsi	Panjang dan tipe data	Null	Jenis Key
Id_event logs_process	Kode Proses Bisnis	Integer (11)	NO	Primary
Id_event logs_client	Kode Klien	Integer (11)	NO	Foreign
Id_source_database	Kode basis data sistem	Integer (11)	NO	Foreign
Process_name	Nama Proses Bisnis	Varchar (255)	NO	

**Tabel 4.14** Atribut untuk event logs activity

Atribut	Deskripsi	Panjang dan tipe data	Null	Jenis Key
Id_event logs_activity	Kode Aktivitas	Integer (11)	NO	Primary
Id_event logs_process	Kode Proses Bisnis	Integer (11)	NO	Foreign
Activity_name	Nama Aktivitas	Varchar (255)	NO	
Activity_order	Urutan Aktivitas	Small integer(6)	NO	

**Tabel 4.15** Atribut untuk tabel event logs source tables

Atribut	Deskripsi	Panjang dan tipe data	Null	Jenis Key
Id_event logs_source_tables	Kode sumber tabel event log	Integer (11)	NO	Primary
Id_event logs_activity	Kode Aktivitas	Integer (11)	NO	Foreign
Id_source_database_table	Kode sumber tabel basis data	Integer (11)	NO	Foreign
Relation_to_column	Kode sumber kolom	Integer (11)	NO	Foreign

	basis data untuk bentuk relasi			
Relation_to_table	Kode sumber kolom basis data untuk bentuk relasi	Integer (11)	NO	Foreign
Order	Urutan untuk pendefinisian tabel utama dan sekunder	Small integer (6)	NO	

**Tabel 4.16** Atribut untuk tabel event logs source columns

Atribut	Deskripsi	Panjang dan tipe data	Null	Jenis Key
Id_event logs_source_column	Kode sumber kolom untuk event logs	Integer (11)	NO	Primary
Id_event logs_activity	Kode Aktivitas	Integer (11)	NO	Foreign
Id_event logs_attr	Kode atribut event log	Integer (11)	NO	Foreign
Id_source_database_column	Kode kolom dari sumber basis data	Integer (11)	NO	Foreign

**Tabel 4.17** Atribut untuk tabel event logs source parameters

Atribut	Deskripsi	Panjang dan tipe data	Null	Jenis Key
Id_event logs_source_parameter	Kode sumber kolom dalam pendefinisian parameter	Integer (11)	NO	Primary
Id_event logs_activity	Kode Aktivitas	Integer (11)	NO	Foreign
Id_source_database_column	Kode kolom untuk sumber basis data	Integer (11)	NO	Foreign
Parameter_value	Nilai parameter	Varchar (100)	NO	

**Tabel 4.18** Atribut untuk tabel source database

Atribut	Deskripsi	Panjang dan	Null	Jenis Key
---------	-----------	-------------	------	-----------

		<b>tipe data</b>		
Id_source_database	Kode basis data	Integer (11)	NO	Primary
Id_event_logs_klien	Kode klien	Integer (11)	NO	Foreign
Id_source_database_vendor	Kode jenis vendor DBMS	Integer (11)	NO	Foreign
System_name	Nama sistem	Varchar (255)	NO	
Database_alias_name	Nama alias basis data	Varchar (255)	NO	
Database_schema	Nama Skema (khusus DBMS PostgreSQL / Oracle	Varchar (255)	YES	
Database_name	Nama basis data	Varchar (255)	NO	
Database_host	Nama host basis data	Varchar (255)	NO	
Database_port	Port yang digunakan basis data	Varchar (255)	NO	
Database_username	Autentifikasi username untuk akses basis data	Varchar (255)	NO	
Database_password	Autentifikasi password untuk akses basis data	Varchar (255)	NO	

**Tabel 4.19** Atribut untuk tabel source database vendor

<b>Atribut</b>	<b>Deskripsi</b>	<b>Panjang dan tipe data</b>	<b>Null</b>	<b>Jenis Key</b>
Id_source_database_vendor	Kode vendor DBMS	Integer (11)	NO	Primary
Vendor_name	Nama vendor DBMS	Varchar (255)	NO	
Vendor_company	Nama perusahaan pemilik vendor DBMS	Varchar (255)	Yes	

**Tabel 4.20** Atribut untuk tabel source database tables

<b>Atribut</b>	<b>Deskripsi</b>	<b>Panjang dan tipe data</b>	<b>Null</b>	<b>Jenis Key</b>
----------------	------------------	----------------------------------	-------------	------------------



Id_source_database_table	Kode tabel basis data	Integer (11)	NO	Primary
Id_source_database	Kode basis data	Integer (11)	NO	Foreign
Table_name	Nama tabel basis data	Varchar (255)	NO	
Table_description	Deskripsi tabel	Varchar (255)	YES	

**Tabel 4.21** Atribut untuk source database columns

Atribut	Deskripsi	Panjang dan tipe data	Null	Jenis Key
Id_source_database_column	Kode kolom untuk tabel basis data	Integer (11)	NO	Primary
Id_source_database_table	Kode tabel basis data	Integer (11)	NO	Foreign
Column_name	Nama kolom tabel	Varchar (255)	NO	
Data_type	Tipe data kolom tabel	Varchar (255)	NO	
Column_length	Panjang tipe data kolom	Integer (11)	NO	
Is_primary_key	Pendefinisian primary key untuk kolom tabel	Small integer (6)	NO	

#### 4.5.2 Perancangan Antarmuka

Pembuatan rancangan antarmuka ini digunakan untuk dasar pembuatan antarmuka aplikasi *Evenl*. Antarmuka ini dibuat berdasarkan aktivitas atau fungsi tiap aktor dalam aplikasi tersebut. Sesuai dengan penjelasan bagian 3.7 pada gambar 3.5 menunjukkan aktivitas yang akan dilakukan tiap aktor aplikasi, dari aktivitas tersebut akan dibuat antarmuka untuk setiap aktivitas dan aktor aplikasi. Rancangan antarmuka akan dibagi berdasarkan aktor sistem, perancangan antarmuka tersebut dijelaskan sebagai berikut

##### 1. Admin *Evenl*

- Pengaturan akses aplikasi

**Form pembuatan akses**

Nama Klien

Jenis Role

Username

Password

Instansi Asal

Alamat klien

Kontak klien

#### Data pengguna aplikasi

▼ Nama Klien	▼ Jenis Role	▼ Instansi Asal
Klien 1	Admin Sistem	Instansi 1
Klien 2	Analisis Bisnis	Instansi 2

**Gambar 4.5.** Form pengaturan akses aplikasi

## 2. Admin Sistem

- Pembuatan akses ke basis data sistem

**Form akses basis data sistem**

Nama Sistem

Analisis Bisnis

Database Vendor

Database Nama

Database Skema

Database Host

Database Port

Database Username

Database Password

**Gambar 4.6** Form pembuatan akses basis data

- Penentuan sumber data *event log*

### Pengaturan sumber tabel & kolom event log

▼ Proses Bisnis & Aktivitas	▼ Sumber tabel dan kolom	▼ Eventlog Atribut
Proses Bisnis - Aktivitas 1	Sumber tabel ▼ Paramter ▼	Atribut kolom ▼
Proses Bisnis - Aktivitas 2	Sumber tabel ▼ Paramter ▼	Atribut kolom ▼
Proses Bisnis - Aktivitas 3	Sumber tabel ▼ Paramter ▼	Atribut kolom ▼

**Gambar 4.7** Form pemetaan informasi event log

### 3. Analisis bisnis

- Ekstraksi event logs dalam bahasa alami

### Ekstraksi event log dengan bahasa alami

▼ Case ID	▼ Activity	▼ Timestamp	▼ Originator
001	Aktivitas 1	2017-12-01 11:22:55	Aktor 1
001	Aktivitas 2	2017-12-01 18:11:23	Aktor 2
001	Aktivitas 3	2017-12-01 19:54:21	Aktor 3
001	Aktivitas 4	2017-12-01 20:44:54	Aktor 4

**Gambar 4.8** Form ekstraksi *event log* dengan bahasa alami

#### 4.1 Uji coba alat ekstraksi

Pada tahap ini akan dilakukan pengujian alat ekstraksi dengan bahasa alami yang sudah ditentukan pada penelitian ini. Sesuai pada penjelasan sebelumnya bahwa pengujian pada penelitian kali ini menggunakan metode *black box testing* dengan menggunakan teknik *Equivalence Partioning*. Yang membagi antara pengujian berdasarkan data yang benar dan pengujian berdasarkan data yang tidak benar. Pengujian ini terdiri dari beberapa scenario yaitu pengujian bahasa alami sebagai masukan oleh analis bisnis yang kemudian ditranslasi ke dalam bahasa

SQL, kemudian pengujian hasil ekstraksi dari hasil CSV yang dikonversi ke dalam format XES dan yang terakhir adalah pengujian hasil XES yang didapat dalam proses konversi yang akan digunakan pada aplikasi ProM.

#### **4.6 Pengujian ekstraksi dan konversi *event logs***

##### **4.6.1 Uji coba ekstraksi event logs dengan bahasa alami**

Pengujian bahasa alami kali ini akan dibagi berdasarkan KPE yang sudah ditentukan dalam tahap sebelumnya kemudian di implementasikan untuk setiap proses bisnis dan aktivitas pada SIM RS dan Odoo ERP. Pada setiap jenis KPE akan diuji pada alat ekstraksi yang dibuat dalam pada penelitian ini. Untuk pengujian data yang benar dalam penelitian ini adalah bahasa alami berdasarkan jenis KPE yang dijelaskan pada bagian 4.2.2 kemudian pengujian data yang tidak benar didasarkan pada batasan frasa proses bisnis dan aktivitas serta operator atribut event logs, operator tanggal, operator bulan dan operator tahun. Hasil yang diharapkan pada penelitian ini yaitu sebuah data event logs sesuai format.

##### **4.6.1.1 Skenario pengujian sesuai kaidah KPE dan batasan token**

Sesuai pada jenis KPE yang sudah ditentukan sebelumnya, setiap jenis KPE akan menggunakan secara random dengan satu contoh kalimat KPE. Seperti pada jenis KPE 1 contoh kalimat perintah bahasa alami yang digunakan yaitu '*Tampilkan distribusi barang*' sehingga akan menampilkan data event logs untuk proses bisnis distribusi barang yang terdiri dari aktivitas permintaan barang, persetujuan barang tampilan kolom yang dihasilkan yaitu berisi informasi atribut *caseID*, *activity*, *timestamp* dan *originator*.

###### **1. Jenis KPE 1**

Kalimat bahasa alami yang digunakan pada pengujian ini yaitu '*tampilkan distribusi barang*'

**Hasil yang diharapkan :** menampilkan semua *event log* untuk proses distribusi barang

###### **2. Jenis KPE 1.1**

Kalimat bahasa alami yang digunakan pada pengujian ini yaitu '*tampilkan distribusi barang bulan januari tahun 2017*'

**Hasil yang diharapkan :** menampilkan semua *event log* untuk proses distribusi barang pada bulan januari tahun 2017

3. Jenis KPE 1.2

Kalimat bahasa alami yang digunakan pada pengujian ini yaitu '*tampilkan distribusi barang pada bulan januari tahun 2017 saring caseID 5600*'

**Hasil yang diharapkan :** menampilkan semua *event log* untuk proses distribusi barang pada bulan januari tahun 2017 dengan tambahan penyaringan atribut *caseID* dengan nilai 5600

4. Jenis KPE 1.3

Kalimat bahasa alami yang digunakan pada pengujian ini yaitu '*tampilkan distribusi barang bulan januari tahun 2017 urutkan berdasarkan caseID*'

**Hasil yang diharapkan :** menampilkan semua *event log* untuk proses distribusi barang pada bulan januari tahun 2017 dengan pengurutan tampilan data berdasarkan atribut *caseID*

5. Jenis KPE 2

Kalimat bahasa alami yang digunakan pada pengujian ini yaitu '*tampilkan Penerimaan Barang*'

**Hasil yang diharapkan :** menampilkan semua aktivitas Penerimaan Barang

6. Jenis KPE 2.1

Kalimat bahasa alami yang digunakan pada pengujian ini yaitu '*tampilkan Penerimaan Barang bulan september tahun 2017*'

**Hasil yang diharapkan :** menampilkan semua aktivitas Penerimaan Barang pada bulan september tahun 2017

7. Jenis KPE 2.2

Kalimat bahasa alami yang digunakan pada pengujian ini yaitu '*tampilkan Penerimaan Barang bulan september tahun 2017 saring caseid 9332*'

**Hasil yang diharapkan :** menampilkan semua aktivitas Penerimaan Barang pada bulan september tahun 2017 dengan tambahan penyaringan data untuk atribut *caseID* yang bernilai 9332'

8. Jenis KPE 2.3

Kalimat bahasa alami yang digunakan pada pengujian ini yaitu '*tampilkan Penerimaan Barang bulan september tahun 2017 urutkan berdasarkan caseid*'

**Hasil yang diharapkan :** menampilkan semua aktivitas Penerimaan Barang pada bulan september tahun 2017 dengan pengurutan data berdasarkan caseID

9. Jenis KPE 3

Kalimat bahasa alami yang digunakan pada pengujian ini yaitu '*tampilkan permintaan barang, persetujuan barang, penerimaan barang*'

**Hasil yang diharapkan :** menampilkan semua aktivitas permintaan barang, persetujuan barang dan penerimaan barang

10. Jenis KPE 3.1

Kalimat bahasa alami yang digunakan pada pengujian ini yaitu '*tampilkan permintaan barang, persetujuan barang, penerimaan barang pada bulan agustus tahun 2017*'

**Hasil yang diharapkan :** menampilkan semua aktivitas permintaan barang, persetujuan barang dan penerimaan barang pada bulan agustus tahun 2017

11. Jenis KPE 3.2

Kalimat bahasa alami yang digunakan pada pengujian ini yaitu '*tampilkan permintaan barang, persetujuan barang, penerimaan barang pada bulan agustus tahun 2017 saring caseid 8756*'

**Hasil yang diharapkan :** menampilkan semua aktivitas permintaan barang, persetujuan barang dan penerimaan barang pada bulan agustus tahun 2017 dengan tambahan penyaringan data berdasarkan atribut *caseID* dengan nilai 8756

12. Jenis KPE 3.3

Kalimat bahasa alami yang digunakan pada pengujian ini yaitu '*tampilkan permintaan barang, persetujuan barang, penerimaan barang pada bulan agustus tahun 2017 urutkan berdasarkan caseID*'

**Hasil yang diharapkan :** menampilkan semua aktivitas permintaan barang, persetujuan barang dan penerimaan barang pada bulan agustus tahun 2017 dengan pengurutan data berdasarkan atribut *caseID*

Dari beberapa contoh kalimat ekstraksi untuk tiap KPE diatas. Pengujian tersebut berhasil memberikan hasil yang sesuai dengan yang diharapkan. Untuk membuktikan hasil tersebut pada gambar 4. Menunjukkan sebuah hasil ekstraksi dengan format event log.

NO	CASE ID	ACTIVITY/TASK	TIMESTAMP	ORIGINATOR
1	6582	Persetujuan Barang	2017-08-07 07:52:40	Oki Fajri Dhuai
2	6582	Penerimaan Barang	2017-08-07 07:54:00	Nur Fauzi Hamid
3	8335	Persetujuan Barang	2017-08-02 12:11:12	Muhammad Fuad Yasin
4	8335	Penerimaan Barang	2017-08-02 12:29:11	Oki Fajri Dhuai
5	8411	Persetujuan Barang	2017-08-04 09:01:53	Oki Fajri Dhuai
6	8411	Penerimaan Barang	2017-08-04 11:35:04	Muhammad Fuad Yasin
7	8639	Persetujuan Barang	2017-08-15 14:18:33	Muhammad Auliz
8	8639	Penerimaan Barang	2017-08-15 14:18:33	Oki Fajri Dhuai
9	8639	Persetujuan Barang	2017-08-09 07:51:58	Nurul Kamariah S
10	8639	Penerimaan Barang	2017-08-09 07:51:58	Oki Fajri Dhuai

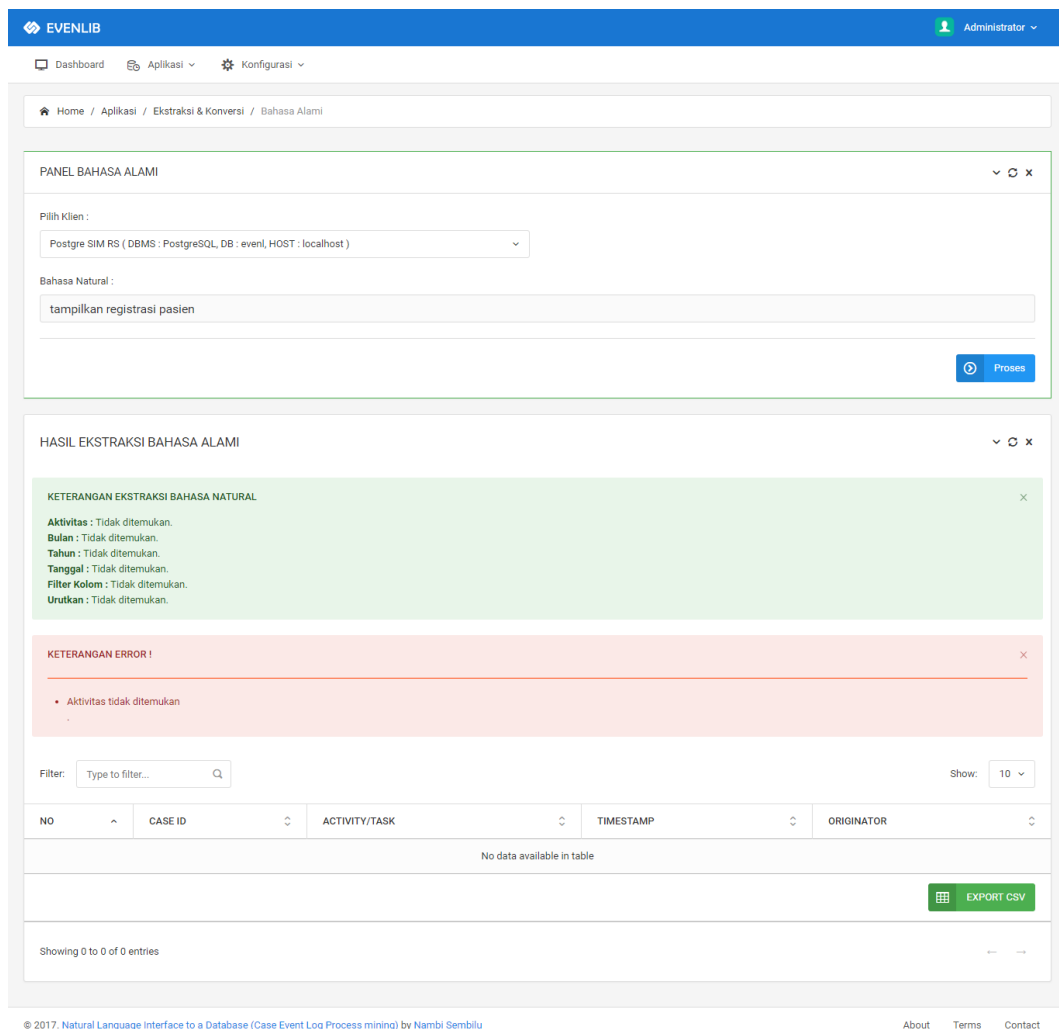
**Gambar 4.9** Contoh salah satu hasil pengujian untuk KPE

#### 4.6.1.2 Skenario pengujian tidak sesuai kaidah KPE dan batasan token

##### 1. Pengujian frasa proses bisnis dan aktivitas

Berdasarkan penjelasan bagian 4.2.2 terdapat beberapa frasa aktivitas dan proses bisnis yang didefinisikan pada penelitian ini. Namun pada pengujian kali ini akan mencoba menggunakan bahasa alami dengan frasa yang tidak didefinisikan dalam penelitian ini. Contoh yang digunakan yaitu dalam Rumah Sakit terdapat proses bisnis bernama *registrasi pasien*, namun pada penelitian proses bisnis tersebut tidak didefinisikan sebagai frasa yang dapat diproses.





**Gambar 4.10** Contoh pengujian tidak sesuai kaidah KPE

## 2. Pengujian operator atribut event log

Dalam penelitian kali ini terdapat perintah bahasa alami ‘*saring*’ yang digunakan untuk penyaringan data dan ‘*urutkan*’ yang digunakan untuk pengurutan data. Perintah tersebut diproses sesuai dengan operator atribut *event log* yang telah didefinisikan. Namun untuk atribut *activity* tidak didefinisikan karena sudah didefinisikan sebagai frasa aktivitas dalam penelitian kali ini. Oleh karena itu pengujian ini akan menggunakan perintah ‘*saring*’ atau ‘*urutkan*’ dengan operator *activity*

Administrator

Dashboard
Aplikasi
Konfigurasi

Home / Aplikasi / Ekstraksi & Konversi / Bahasa Alami

PANEL BAHASA ALAMI

Pilih Klien :  
Postgre SIM RS ( DBMS : PostgreSQL, DB : evenli, HOST : localhost )

Bahasa Natural :  
tampilkan distribusi barang urutkan berdasarkan activity

Proses

HASIL EKSTRAKSI BAHASA ALAMI

KETERANGAN EKSTRAKSI BAHASA NATURAL
Aktivitas : permintaan barang, persetujuan barang, penerimaan barang  
Bulan : Tidak ditemukan.  
Tahun : Tidak ditemukan.  
Tanggal : Tidak ditemukan.  
Filter Kolom : Tidak ditemukan.  
Urutkan : Tidak ditemukan.

Filter: Type to filter...
Show: 10

NO	CASE ID	ACTIVITY/TASK	TIMESTAMP	ORIGINATOR
1	6153	Permintaan Barang	2017-02-03 15:46:12	Adi Wasis Prakosa
2	7398	Permintaan Barang	2017-04-26 14:25:33	Adi Wasis Prakosa
3	7138	Permintaan Barang	2017-04-12 09:42:03	Adi Wasis Prakosa
4	6918	Permintaan Barang	2017-03-27 17:13:29	Adi Wasis Prakosa
5	6886	Permintaan Barang	2017-03-24 11:24:17	Adi Wasis Prakosa
6	3735	Permintaan Barang	2016-08-05 13:45:58	Budi Suprpti
6	3735	Permintaan Barang	2016-08-05 13:45:58	Budi Suprpti
7	2630	Permintaan Barang	2016-05-18 10:17:19	TOETIK ARYANI
8	2540	Permintaan Barang	2016-05-12 20:41:46	TOETIK ARYANI
9	3794	Permintaan Barang	2016-08-10 13:02:20	TOETIK ARYANI
10	3021	Permintaan Barang	2016-06-13 07:58:40	TOETIK ARYANI

EXPORT CSV

Showing 1 to 10 of 23,176 entries

1 2 3 4 5 ... 2318

© 2017. Natural Language Interface to a Database (Case Event Log Process mining) by Nambi Sembilu

About Terms Contact

**Gambar 4.11** Contoh pengujian atribut *event log*

### 3. Pengujian operator tanggal

Batasan operator tanggal yang digunakan yaitu format tanggal 01-31 akan tetapi pada bulan terdapat beberapa format tanggal yang diterima sistem tapi secara kenyataan tidak benar contohnya 30 februari tahun 2017 karena batasan maksimal tanggal untuk februari tahun 2017 adalah 28. Selanjutnya pengujian ini menggunakan periode waktu 01 februari 2017 sampai dengan 30 februari tahun 2017



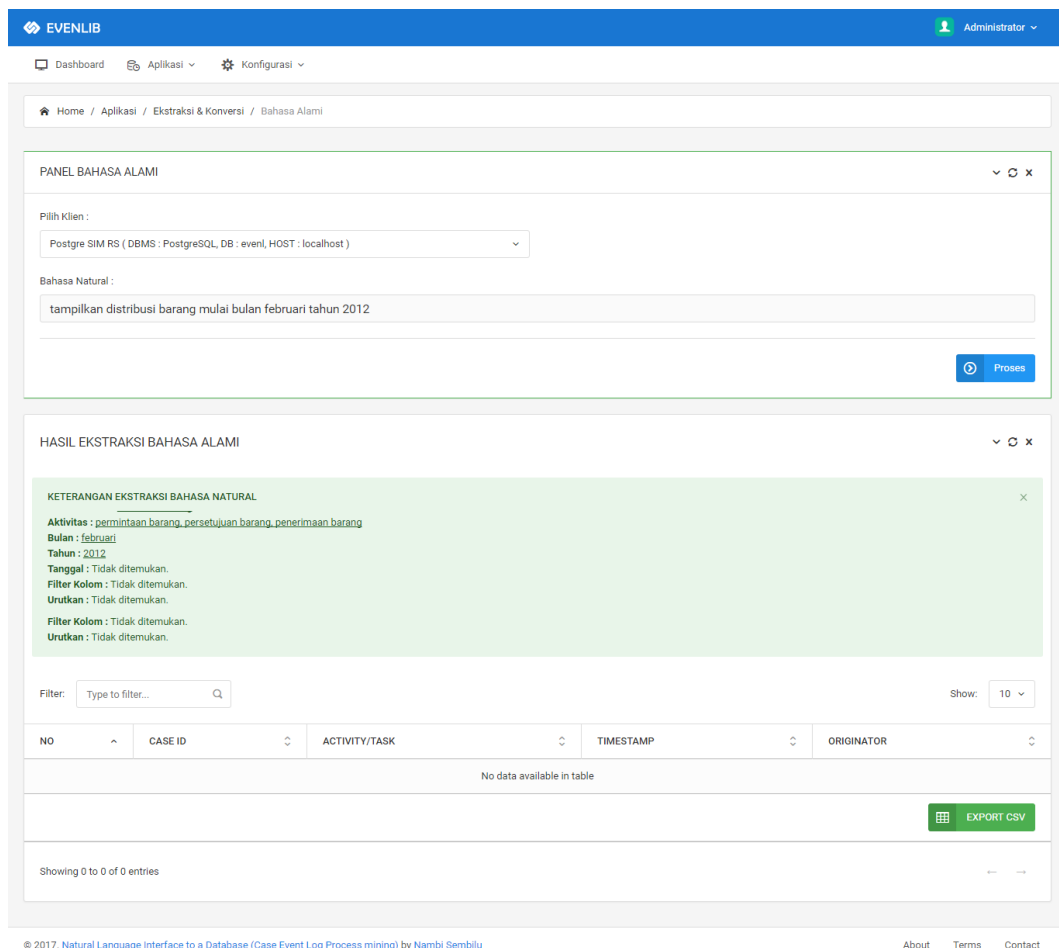
terjadi saat pada bahasa alami. Contoh kesalahan penulisan adalah bulan ‘febuari’

The screenshot displays the EVENLIB web application interface. At the top, there is a blue header with the logo and navigation links: Dashboard, Aplikasi, and Konfigurasi. The user is logged in as Administrator. The main content area is divided into two panels. The top panel, titled 'PANEL BAHASA ALAMI', contains a dropdown menu for 'Pilih Klien' (Postgre SIM RS) and a text input field for 'Bahasa Natural' containing the text 'tampilkan distribusi barang bulan febuari'. A 'Proses' button is located at the bottom right of this panel. The bottom panel, titled 'HASIL EKSTRAKSI BAHASA ALAMI', shows the results of the extraction. It includes a green box with 'KETERANGAN EKSTRAKSI BAHASA NATURAL' listing activities like 'permintaan barang' and 'persetujuan barang'. Below this is a red box with 'KETERANGAN ERROR !' listing two errors related to the month and year format. At the bottom, there is a table with columns: NO, CASE ID, ACTIVITY/TASK, TIMESTAMP, and ORIGINATOR. The table is currently empty, showing 'No data available in table'. A filter input and a 'Show: 10' dropdown are also present. An 'EXPORT CSV' button is located at the bottom right of the table area.

**Gambar 4.13** Contoh operator bulan

## 5. Pengujian operator tahun

Batasan pada penelitian ini untuk operator tahun adalah 4 digit angka mulai 1900 sampai dengan tahun berjalan. Akan tetapi data yang didapatkan pada penelitian ini hanya sebatas tahun 2014-2017 untuk SIM RS dan tahun 2018 pada OdooERP. Maka pengujian kali ini akan mencoba tahun 2012 sebagai input operator tahun pada bahasa alami.



**Gambar 4.14** Contoh pengujian operator tahun

#### 4.6.2 Uji coba konversi format XES

Pada tahapan pengujian ini, akan hasil data yang berbentuk CSV yang kemudian dimasukan ke aplikasi ProM Import, Disco dan aplikasi konversi berbasis web pada halaman <http://www.satcos.in/processmining/index.php> yang kemudian akan menghasilkan data dengan format *event log* XES. Satu data event log terdiri dari kumpulan beberapa instance atau *caseID* yang memiliki beberapa aktivitas yangurut sesuai dengan waktu kejadian. Pada penelitian ini data yang digunakan adalah sistem SIM RS dengan proses bisnis distribusi barang dan pemberian e-resep serta sistem Odoo ERP dengan proses bisnis order pembelian dan order penjualan.

Pengujian data yang sesuai format event log atau dalam arti data yang dihasilkan memiliki CaseID dengan beberapa aktivitas yangurut berdasarkan penentuan proses bisnis, dari hasil ekstraksi tersebut akan dicoba digunakan dalam

aplikasi Disco dan aplikasi berbasis web yang digunakan untuk mengkonversi format CSV menjadi XES. Untuk memasukan format CSV ke dalam kedua aplikasi tersebut, perlu dipilih tiap atribut *event log* yang disesuaikan dengan header kolom CSV yang ada.

**SATCOS** HOME / PROGRAMMING HARDWARE MICROCONTROLLER SITEMAP

## ONLINE XES CONVERTER

[Quick Tutorial on "How to use this tool?"](#)

Event Log in table format:

Choose File eventlib-csv-01262018175439.csv

Advanced Options:

☐ OFF

Row ID	case_id	activity	timestmp	originator
0	5431	Permintaan Barang	2017-01-09 09:05:05	Nur Fauzi Hamidi
1	5586	Permintaan Barang	2017-01-03 12:29:04	Nur Fauzi Hamidi
2	5594	Permintaan Barang	2017-01-03 23:19:07	Nur Fauzi Hamidi
3	5596	Permintaan Barang	2017-01-01 19:36:19	Erca Tri Herliwanti
4	5597	Permintaan Barang	2017-01-02 05:08:27	Tirsa Rizky Ersanti
5	5598	Permintaan Barang	2017-01-02 11:44:44	Febriansyah Nur Utomo
6	5599	Permintaan Barang	2017-01-03 07:58:40	Nur Fauzi Hamidi
7	5600	Permintaan Barang	2017-01-03 07:58:49	Nur Fauzi Hamidi
8	5601	Permintaan Barang	2017-01-03 08:04:26	Nur Fauzi Hamidi
9	5602	Permintaan Barang	2017-01-03 08:29:11	Ajeng Widya Utami

Project Title:

SIM RS - Aktivitas Permintaan Barang

Format: ☒ Simple ☐ Advanced

Case Name/Id: case\_id

Event Name: activity

Event ID: None

Event Start Time: None

Event End Time: timestmp YYYY-MM-DD HH:mm:ss

Life Cycle Stage: None

Event Time: None

Resource Name/ID: originator

Resource Role: None

Resource Group: None

Start Conversion

**Gambar 4.15** Konfigurasi atribut Event log dengan masukan dataCSV

Kemudian proses pengujian kali ini didapatkan bahwa hasil ekstraksi yang ada pada penelitian ini berhasil dikonversi ke dalam format CSV tanpa kendala.

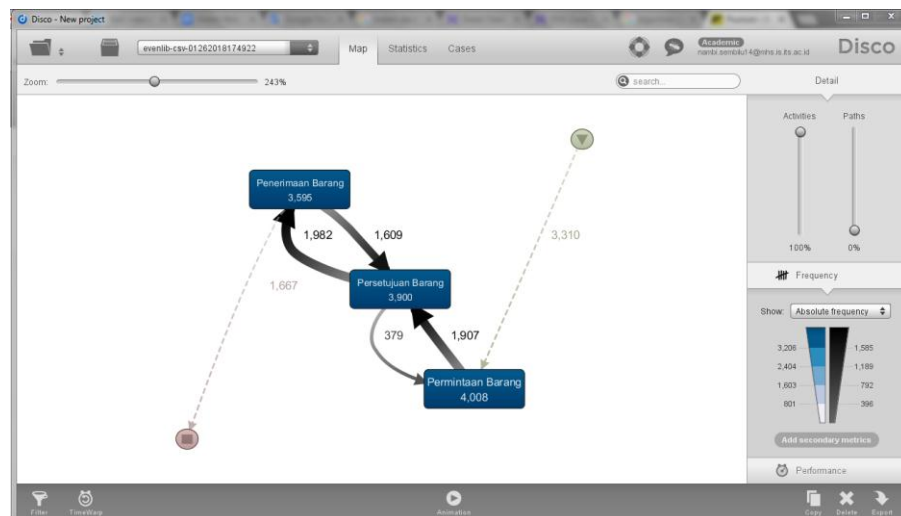
```
XesGeneration::Info::Started conversion process...
XesGeneration::Info::Successfully gathered parsing parameters.
XesGeneration::Info::Parsing the complete file...
XesGeneration::Info::11583 records parsed in the data file.
XesGeneration::Info::Successfully gathered xes parameters.
XesGeneration::Info::Successfully gathered xes setting.
XesGeneration::Info::Preprocessing the data...
XesGeneration::Info::Successfully preprocessed data.
XesGeneration::Info::Generating tags...
XesGeneration::Info::4013 traces tags generated
XesGeneration::Info::11583 event tags generated
```

**Gambar 4.16** Hasil konversi dengan *tools* satcos

#### 4.6.3 Uji coba process mining dengan ProM

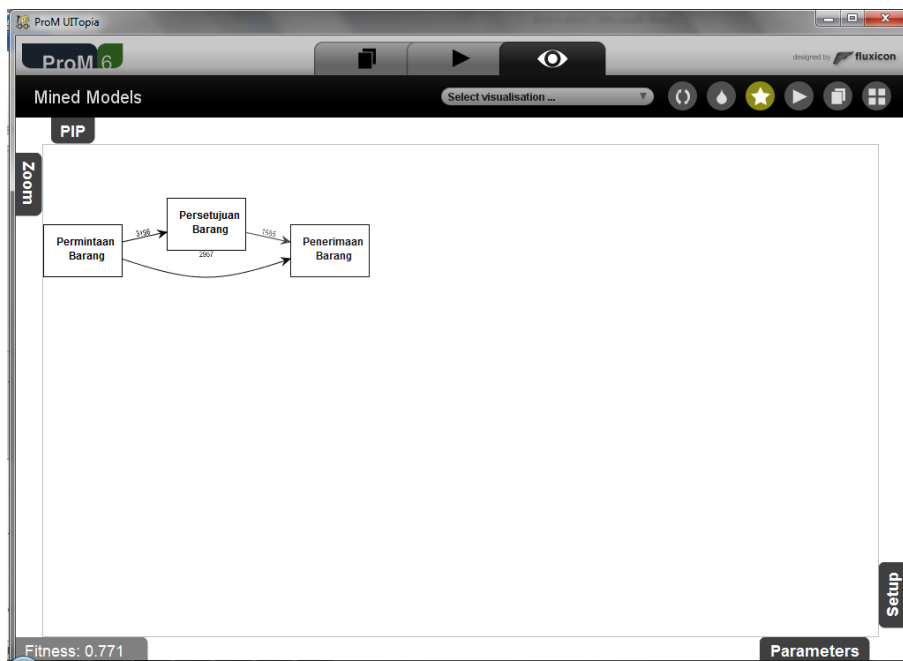
Untuk proses pengujian pada terakhir ini hasil *event log benar* yang didapatkan dari proses konversi ke format XES kemudian akan dimasukkan ke dalam aplikasi bernama ProM ataupun Disco yang bertujuan untuk kepentingan analisa. Dalam analisa *process mining* dibutuhkan sebuah algoritma tertentu agar mendapatkan hasil yang diharapkan. Pada penelitian pengujian dilakukan dengan menggunakan algoritma *heuristic miner* dan *inductive miner* pada aplikasi ProM dan algoritma fuzzy pada aplikasi Disco. Data *event log* yang digunakan pada pengujian kali ini yaitu data SIM-RS pada proses bisnis distribusi barang pada tahun 2017.

- Hasil *import* format XES dengan aplikasi Disco



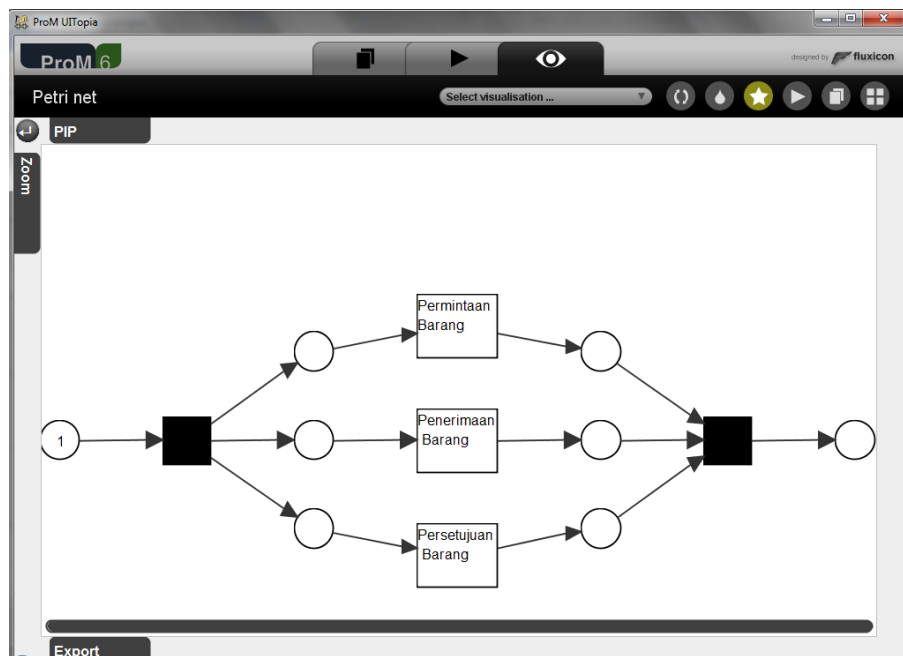
**Gambar 4.17** Hasil pengujian format XES aplikasi Disco

- Hasil *import* format XES dengan aplikasi ProM
  - Algoritma *heuristic miner*



**Gambar 4.18** Hasil pengujian heuristic miner aplikasi ProM

- Algoritma *inductive miner*



**Gambar 4.19** Hasil pengujian inductive miner aplikasi ProM



#### 4.6.4 Uji akurasi ekstraksi event log dengan bahasa alami

Kurva ROC adalah tool dua dimensi yang digunakan untuk menilai kinerja klasifikasi yang menggunakan dua class keputusan, masing-masing objek dipetakan ke salah satu elemen dari himpunan pasangan, positif atau negatif. Pada kurva ROC, TP rate diplot pada sumbu Y dan FP rate diplot pada sumbu X. Untuk klasifikasi data mining, nilai AUC dapat dibagi menjadi beberapa kelompok (Gorunescu, 2011).

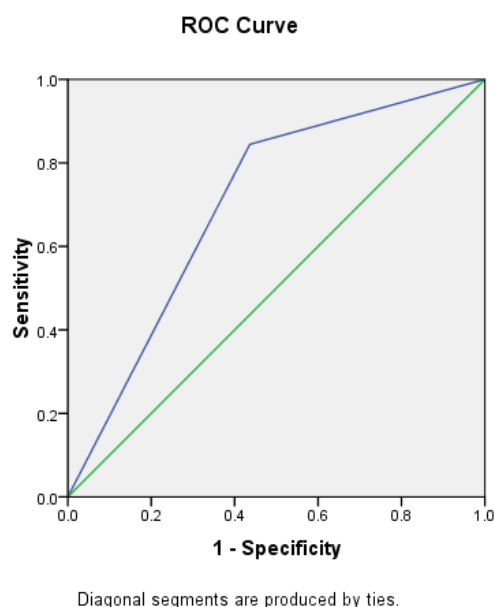
- 0.90-1.00 = *Excellent Classification*
- 0.80-0.90 = *Good Classification*
- 0.70-0.80 = *Fair Classification*
- 0.60-0.70 = *Poor Classification*
- 0.50-0.60 = *Failure*

Kurva ini banyak digunakan untuk menilai hasil prediksi, kurva ROC adalah teknik untuk memvisualisasikan, mengatur, dan memilih pengklasifikasian berdasarkan kinerja mereka (Gorunescu, 2011). Pada penelitian ini untuk memastikan bahwa aplikasi ini sudah layak untuk digunakan perlu dilakukan maka perlu dilakukan pengukuran tingkat akurasi aplikasi ekstraksi *event log* pada penelitian ini maka akan dilakukan pengujian sejumlah 100 kalimat berdasarkan KPE yang ada kemudian akan diselaraskan dengan kebutuhan seorang analis bisnis untuk menunjukkan validitas hasil ekstraksi event log. Dari 100 kalimat tersebut akan ditunjukkan hasil uji aplikasi dan dibandingkan dengan hasil validitas dari seorang analis bisnis. Hal tersebut ditujukan untuk mengetahui bahwa aplikasi ini benar-benar dapat digunakan untuk proses analisa dalam bidang proses mining. Untuk melihat hasil pengujian 100 kalimat beserta validitas hasilnya dapat dilihat pada lampiran 2. Perhitungan yang digunakan adalah *precision* dan *recall*. Dua pengukuran tersebut digunakan untuk mengetahui sejauh mana kinerja aplikasi dan metode yang digunakan dalam penelitian ini. Kemudian untuk dari pengujian tersebut didapatkan hasil sebagai berikut.

**Tabel 4.22** Tabel pengujian *True False*

		Nilai Sebenarnya	
		PSOTIVE	FALSE
Nilai Prediksi	TRUE	38	24
	FALSE	7	31

Dari tabel diatas maka perhitungan didapatkan nilai precision adalah 0.612 atau 61% sedangkan untuk nilai *recall* adalah 0.550 atau 50 %. Kemudian untuk nilai akurasi didapatkan dari tabel 4.22 yang didapatkan yaitu 0.8444 atau 84 %.. Setelah itu untuk memperlihatkan kinerja algoritma ekstrasi event log bentuk grafik pada penelitian ini menggunakan *Reciever Operating Characteristic* (ROC). Software yang digunakan memproses kurva ROC adalah SPSS dengan memasukan hasil prediksi dengan hasil sebenarnya. Dari perhitungan tersebut didapatkan kurva ROC sebagai berikut :

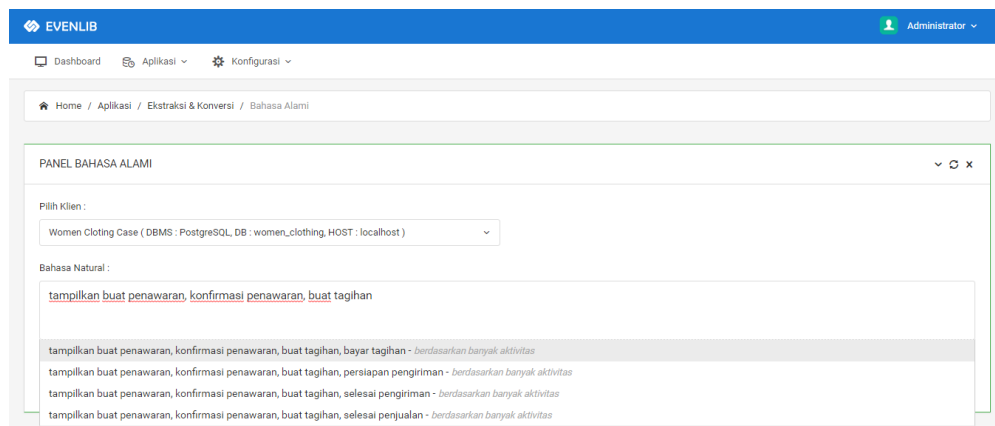


**Gambar 4.20** kurva ROC untuk ekstraksi event log

Dari hasil kurva tersebut dapat dijelaskan bahwa nilai optimal untuk sensitivitas sebesar 0.844 atau 84 % sedangkan nilai optimal spesifisitas sebesar 0.436 atau 44% sedangkan untuk nilai rata-rata kurva ROC adalah 0.704 atau 70 %. Beberapa kesalahan yang didapatkan dari pengujian akurasi dikarenakan

kesalahan ejaan ataupun kesalahan struktur kalimat dalam bahasa alami yang sudah ditentukan sebelumnya.

Untuk menutupi kekurangan kesalahan penulisan struktur kalimat pada penelitian ini menambahkan bentuk saran yang terstruktur sesuai pada kaidah kalimat perintah ekstraksi pada antarmuka pengguna saat proses penulisan bahasa



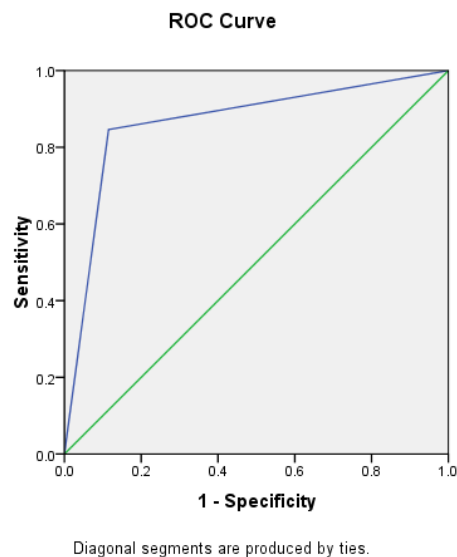
**Gambar 4.21** Contoh bentuk otomatis saran

Kemudian untuk kekurangan pada penulisan ejaan frasa proses bisnis dan aktivitas pada bahasa alami, pada penelitian ini menambahkan bentuk rekomendasi perbaikan frasa dalam antarmuka dengan saran penulisan yang benar

Pada fungsi rekomendasi perbaikan farasa dan aktivitas digunakan algoritma (Oliver, 1994) atau dikenal dengan fungsi `similar_text` pada bahasa pemrograman PHP dibandingkan dengan metode *Longest Common Subsequence* (LCS) oleh (Paterson dan Dančák, 1994). Kedua algoritma tersebut akan diuji untuk dapat menilai kinerja algoritma tersebut dalam memperbaiki frasa dalam penggunaan penulisan bahasa alami untuk ekstraksi event logs. Sebelum proses pengujian akurasi kedua algoritma tersebut, pada penelitian ini melakukan proses pengujian awal untuk mendapatkan nilai minimal kemiripan dalam perbaikan frasa proses bisnis dan aktivitas. Dari proses awal tersebut didapatkan bahwa nilai minimal yang optimal untuk algoritma pertama yaitu 86% dan untuk algoritma kedua yaitu 93%. Hasil tersebut didapatkan dengan mengambil nilai rata-rata pada nilai persentase pada pengujian beberapa kesalahan penulisan frasa dengan frasa yang sebenarnya. Dikatakan nilai optimal dengan asumsi bahwa kesesuaian antara kesalahan penulisan frasa dengan frasa sebenarnya dimaksud, jika nilai tersebut diperbesar akan mengurangi kemungkinan banyaknya rekomendasi frasa yang

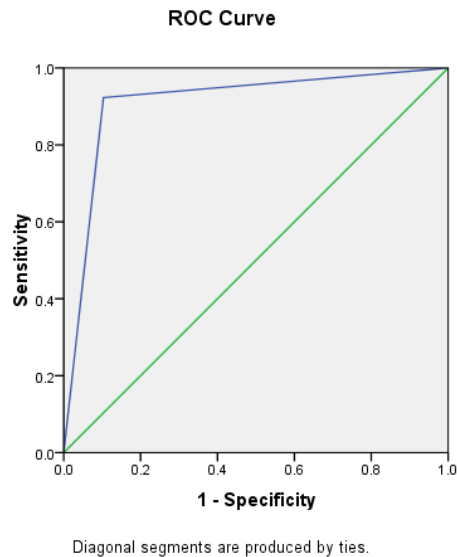
dimaksud, sehingga memperkecil kemungkinan dalam mendapatkan rekomendasi frasa yang sesuai. Begitupun jika nilai optimal tersebut diperkecil akan memperbanyak rekomendasi frasa yang dimaksud, sehingga kemungkinan hasil yang diberikan tidak sesuai dengan frasa yang dimaksud dalam penelitian ini. Setelah didapatkan nilai minimal yang optimal selanjutnya akan diuji kinerja kedua algoritma tersebut. Dalam pengujian kinerja disediakan sebanyak 100 frasa proses bisnis dan aktivitas pada algoritma.

Pada algoritma pertama diuji dengan menggunakan kurva ROC dengan hasil nilai sensitivitas sebesar 0.846 atau 85%, nilai spesifisitas sebesar 0.115 atau 11 % dan nilai kurva ROC rata-rata sebesar 0.866 atau 87 %.



**Gambar 4.22** kurva ROC algoritma php similar\_text

Kemudian untuk algoritma kedua didapatkan hasil nilai sensitivitas sebesar 0.923 atau 92 % dan spesifitas sebesar 0.103 atau 10 % dengan nilai kurva ROC rata-rata sebesar 0.910 atau 91 %.



**Gambar 4.23** kurva ROC algoritma LCS

Dengan membandingkan kedua algoritma tersebut didapatkan bahwa algoritma LCS memberikan hasil yang terbaik dalam perbaikan frasa proses bisnis atau aktivitas saat penulisan bahasa alami untuk proses ekstraksi event log. Hasil pengujian pada tahap ini dapat dilihat pada lampiran 4

**Gambar 4.24** Contoh otomatis perbaikan bahasa alami

Dengan tambahan antarmuka otomatis saran dan otomatis perbaikan pada penelitian ini maka dapat mengurangi tingkat kesalahan pada saat penulisan bahasa alami.

#### 4.6.5 User Acceptance Test (UAT) oleh Analis Bisnis

Pada terakhir dalam pengujian ini akan melibatkan seorang Analis Bisnis yang untuk menguji kesesuaian hasil format *event log* yang ditampilkan dengan bahasa alami masukan. Pengujian ini dilakukan agar alat ekstraksi ini dapat digunakan sesuai kebutuhan analis bisnis. Pakar yang dilibatkan dalam penelitian ini adalah “Mahendrawathi ER, ST., MSc., PhD.” yang merupakan akademisi dan praktisi analis bisnis dalam bidang bisnis proses manajemen dengan spesialisasi pengetahuan dalam Odoo ERP. Hasil UAT sudah dilakukan pada tanggal 15 Maret 2018 dari 4 poin yang diujikan, semua poin tersebut berhasil sesuai dengan hasil yang diharapkan.

#### 4.7 Analisa Hasil

Dari beberapa proses pada penelitian dan telah dilakukan pengujian beberapa hasil dan temuan yang telah didapatkan. Pada pengujian studi kasus SIM Rumah Sakiit pada proses bisnis distribusi barang sumber informasi *event log* tersimpan pada beberapa tabel basis data, pendefinisian atribut event log didasarkan pada relasi dari tabel transaksi *farmasi\_permintaan* dengan melibatkan tabel acuan *farmasi\_permintaan\_unit* dan tabel *pengguna*.

id_farmasi_permi	id_farmasi_br	id_farmasi_permi	jumlah_bar	status_distrib	jumlah_distrib	id_farmasi_nama
43	7441	20	100	1	100	(Null)
46	7340	21	5	1	5	(Null)
47	7464	21	1	1	1	(Null)
48	6658	21	1	1	1	(Null)
49	7598	22	50	1	50	(Null)
51	7603	22	20	0	0	(Null)
52	7124	22	1	1	1	(Null)
55	6835	22	3	1	3	(Null)
60	7512	23	1	1	1	(Null)
61	7547	23	1	1	1	(Null)
62	6658	23	1	1	1	(Null)
63	6657	23	1	1	1	(Null)
64	6834	23	4	1	4	(Null)
75	6865	27	1	1	1	(Null)
77	7109	27	1	1	1	(Null)
78	7614	27	45	1	45	(Null)

**Gambar 4.25** Bentuk isi Tabel *farmasi\_permintaan*

id_farmasi	id_gudang	id_petugas	waktu_permintaan	status	waktu_persetujuan	id_petugas	id_petugas	waktu_penerimaan
10093	12	232016	2017-10-16 15:28:43	1	2017-10-16 15:28:55	233403	232016	2017-10-16 15:36:02
10092	12	232016	2017-10-16 13:51:13	1	2017-10-16 14:03:56	233404	232016	2017-10-16 14:26:57
10091	12	232016	2017-10-16 13:34:28	1	2017-10-16 13:52:36	233404	232016	2017-10-16 15:18:04
10090	6	232031	2017-10-16 12:27:22	1	2017-10-16 12:49:09	233404	232014	2017-10-16 13:11:42
10089	12	232016	2017-10-16 12:38:05	1	2017-10-16 12:58:00	233403	232016	2017-10-16 13:07:54
10088	90	182628	2017-10-16 12:16:48	0	(Null)	(Null)	(Null)	(Null)
10087	7	232029	2017-10-16 11:20:55	0	(Null)	(Null)	(Null)	(Null)
10086	126	182672	2017-10-16 13:08:33	0	(Null)	(Null)	(Null)	(Null)
10085	6	232031	2017-10-16 11:19:35	0	(Null)	(Null)	(Null)	(Null)
10084	18	182665	2017-10-16 10:01:58	0	(Null)	(Null)	(Null)	(Null)
10083	19	182665	2017-10-16 09:59:04	0	(Null)	(Null)	(Null)	(Null)
10082	25	182665	2017-10-16 09:56:41	0	(Null)	(Null)	(Null)	(Null)
10080	26	182665	2017-10-16 09:55:20	0	(Null)	(Null)	(Null)	(Null)
10079	7	232029	2017-10-16 09:51:18	1	2017-10-16 09:55:47	233403	232029	2017-10-16 15:11:58
10078	6	182631	2017-10-16 09:38:13	1	2017-10-16 13:04:47	233403	232014	2017-10-16 13:12:07
10077	89	135576	2017-10-16 09:23:10	0	(Null)	(Null)	(Null)	(Null)

**Gambar 4.26** Bentuk isi Tabel farmasi\_permintaan\_unit

	nm_pengguna	tgl_lahir_pengg	kelamin	id_tempat	id_rogelar	gelar	nm_asli
2	MOKHAMMAD FA	1989-11-16 00:00:00	1	SURABAYA	17		
7	SUGENG KURNIA	1983-03-21 00:00:00	1		22	S.Si.	SUGENG KURNIA
16	IKA ANITA SARI	1988-04-30 00:00:00	2	JOMBANG	28		IKA ANITA SARI
17	DEASY WULAN SARI	1987-12-20 00:00:00	1	SURABAYA	3		DEASY WULAN SARI
18	VIKI HARDANI	1988-02-24 00:00:00	2	MALANG	3		VIKI HARDANI
19	IFADA CHAIRANIS	1989-04-15 00:00:00	2	BENGKULU	3		IFADA CHAIRANIS
20	ANDHITA LAKSMI	1988-03-04 00:00:00	1	JAKARTA	3		ANDHITA LAKSMI
21	HANDY MUBARO	1988-06-01 00:00:00	1	KEDIRI	28		HANDY MUBARO
22	ODIYANTO ADI PR	1988-06-15 00:00:00	1	N	3		ODIYANTO ADI PR
23	ICHSANNUDHIN	1987-10-31 00:00:00	1	SURABAYA	28		ICHSANNUDHIN
24	CATUR PRIYO UT	1987-12-07 00:00:00	1	N	3		CATUR PRIYO UT
25	RIZAL HANIEF MA	1988-04-29 00:00:00	1	KEDIRI	28		RIZAL HANIEF MA
26	AKHDHAN MAULI	1984-06-28 00:00:00	1	SUMENEP	3		Akhdhan Ma
27	HENRY I YUDONC	1988-06-13 00:00:00	2	JAKARTA	3		HENRY I YUDONC
28	NESSYA WIDASA	1988-12-15 00:00:00	2	SURABAYA	28		NESSYA WIDASA
29	GIVI HERDIANDA	1989-06-16 00:00:00	2	SURABAYA	28		GIVI HERDIANDA

**Gambar 4.27** Bentuk isi Tabel pengguna

Dari ketiga tabel tersebut sudah berhasil didapatkan *event log* untuk aktivitas “Permintaan Barang”, “Persetujuan Permintaan” dan “Penerimaan barang” yang ditandai dengan kolom *waktu\_permintaan*, *waktu\_persetujuan* dan *waktu\_penerimaan* sebagai atribut timestamp, kemudian untuk atribut *originator* diambil dari kolom *id\_petugas\_permintaan*, *id\_petugas\_persetujuan* dan *id\_petugas\_penerima* yang berelasi dengan tabel pengguna.

pg-localhost even public Run Stop Explain

```

1 SELECT * FROM (SELECT * FROM ( SELECT farmasi_permintaan_unit.id farmasi_permintaan_unit as case_id, 'Permintaan Barang' as activity, to_char(
farmasi_permintaan_unit.waktu_permintaan, 'YYYY-MM-DD HH24:MI:SS') as timestamp, pengguna.nm_pengguna as originator FROM
farmasi_permintaan_unit JOIN pengguna on trim(pengguna.id_pengguna::text)=trim(farmasi_permintaan_unit.id_petugas_peminta::text) WHERE
farmasi_permintaan_unit.waktu_permintaan IS NOT NULL AND farmasi_permintaan_unit.waktu_permintaan BETWEEN '2017-05-01' AND '2017-07-30'
group by case_id, activity, timestamp, originator ) AS A1 UNION ALL ( SELECT farmasi_permintaan_unit.id farmasi_permintaan_unit as case_id,
'Persetujuan Permintaan' as activity, to_char(farmasi_permintaan_unit.waktu_persetujuan, 'YYYY-MM-DD HH24:MI:SS') as timestamp, pengguna.
nm_pengguna as originator FROM farmasi_permintaan_unit JOIN pengguna on trim(pengguna.id_pengguna::text)=trim(farmasi_permintaan_unit.
id_petugas_persetujuan::text) WHERE farmasi_permintaan_unit.waktu_persetujuan IS NOT NULL AND farmasi_permintaan_unit.waktu_persetujuan
BETWEEN '2017-05-01' AND '2017-07-30' group by case_id, activity, timestamp, originator ) UNION ALL ( SELECT farmasi_permintaan_unit.
id farmasi_permintaan_unit as case_id, 'Penerimaan Barang' as activity, to_char(farmasi_permintaan_unit.waktu_penerimaan, 'YYYY-MM-DD
HH24:MI:SS') as timestamp, pengguna.nm_pengguna as originator FROM farmasi_permintaan_unit JOIN pengguna on trim(pengguna.id_pengguna::text)=
trim(farmasi_permintaan_unit.id_petugas_penerima::text) WHERE farmasi_permintaan_unit.waktu_penerimaan IS NOT NULL AND
farmasi_permintaan_unit.waktu_penerimaan BETWEEN '2017-05-01' AND '2017-07-30' group by case_id, activity, timestamp, originator ) ) AS B ORDER
BY timestamp asc

```

Message Result 1

case_id	activity	timestamp	originator
7446	Penerimaan	2017-05-01	(Anisa Dwi Syafitri
7446	Persetujuan	2017-05-01	(Muhammad Fuad
7455	Permintaan	2017-05-01	(Nurul Kamariyah
7456	Permintaan	2017-05-01	(Muhammad Syam
7457	Permintaan	2017-05-01	MARIA ULFA
7458	Permintaan	2017-05-01	Ratna Nisa' Anggr

SELECT \* FROM (SELECT \* FROM ( SELECT farmasi\_permintaan\_unit.id farmasi\_permintaan\_unit as case\_id, ' Read Only Query time: 4.269s Record 1 of 3366

**Gambar 4.28** hasil ekstraksi sistem Rumah Sakit dengan SQL

Pilih Klien :  
Postgre SIM RS (DBMS : PostgreSQL, DB : evenl, HOST : localhost)

tampilkan proses distribusi barang mulai tanggal 1 mei 2017 sampai tanggal 30 juli 2017

Proses

HASIL EKSTRAKSI BAHASA ALAMI

Waktu eksekusi program 0.07405 detik

KETERANGAN EKSTRAKSI BAHASA NATURAL

Proses Bisnis : distribusi barang  
Aktivitas : permintaan barang, persetujuan permintaan, penerimaan barang  
Bulan : Tidak ditemukan.  
Tahun : Tidak ditemukan.  
Tanggal : 1 mei 2017, 30 juli 2017  
Filter Kolom : Tidak ditemukan.  
Urutkan : Tidak ditemukan.

Filter: Type to filter... Show: 10

NO	CASE ID	ACTIVITY/TASK	TIMESTAMP	ORIGINATOR
1	7446	Penerimaan Barang	2017-05-01 04:45:15	Anisa Dwi Syafitri
2	7446	Persetujuan Permintaan	2017-05-01 04:45:15	Muhammad Fuad Yasin
3	7455	Permintaan Barang	2017-05-01 08:24:38	Nurul Kamariyah S
4	7456	Permintaan Barang	2017-05-01 08:44:49	Muhammad Syamsul Hidayat
5	7457	Permintaan Barang	2017-05-01 11:03:21	MARIA ULFA
6	7458	Permintaan Barang	2017-05-01 13:04:01	Ratna Nisa' Anggraini
7	7459	Permintaan Barang	2017-05-01 13:41:48	Farida Fuji Astutik
8	7460	Permintaan Barang	2017-05-01 18:13:19	Ratna Nisa' Anggraini
9	7461	Permintaan Barang	2017-05-01 18:21:07	Ratna Nisa' Anggraini
10	7462	Permintaan Barang	2017-05-02 08:16:58	Anisa Adyalina

EXPORT CSV

Showing 1 to 10 of 3,366 entries

1 2 3 4 5 ... 337

**Gambar 4.29** hasil ekstraksi sistem Rumah Sakit dengan NLIDB



Dari hasil ekstraksi kemudian akan dibandingkan dengan jumlah data awal pada tiap tabel pada basis data sistem Rumah Sakit.

**Tabel 4.23** Perbandingan data sistem Rumah Sakit

SUMBER DATA	SEBELUM DI EKTRAK		SETELAH DI EKSTRAK		
	TABEL	$\Sigma$ RECORD	KETERANGAN	$\Sigma$ SQL	$\Sigma$ NLIDB
SITEM RUMAH SAKIT	farmasi_permintaan	82,122	Mulai tanggal 1 mei 2017 sampai 31 mei 2017	3,366	3,366
	farmasi_permintaan_unit	9,074			
	pengguna	177,324			

Pada tabel diatas dapat dilihat jumlah data pada basis data sistem Rumah Sakit untuk tabel terkait sebelum dan sesudah dilakukan ekstraksi untuk kebutuhan *event logs*. Dari hasil tersebut dapat menunjukkan bahwa konsep ekstraksi data *event logs* dalam penelitian ini sudah berhasil dijalankan.

Untuk pengujian studi kasus proses bisnis order penjualan pada sistem Odoo ERP, sumber informasi *event log* yang tersimpan lebih kompleks dibandingkan dengan SIM Rumah Sakit. Hal ini didasarkan karena Odoo ERP tidak secara langsung menyimpan informasi tiap aktivitas dalam sistem pada tabel transaksi mereka. Oleh karena itu pada penelitian ini memanfaatkan fitur *message\_log* yang tersedia pada Odoo ERP. Tabel yang digunakan pada fitur tersebut adalah tabel yang memiliki nama awalan *mail*, namun pada kasus pengambilan informasi event log pada penelitian ini hanya memanfaatkan data yang tersimpan pada tabel *mail\_message* dan *mail\_tracking\_value* untuk mendapatkan informasi atribut *activity* dan atribut *timestamp*. Selain itu tabel lain yang dilibatkan yaitu tabel *res\_users*, *res\_partner* untuk membantu mendapatkan atribut originator, kemudian untuk mendapatkan atribut *case id* perlu melibatkan tabel-tabel transaksi pada Odoo ERP.

id	write_uid	create_uid	res_id	date	model	message_type
1	1	1	1	2018-02-17 06:37:30	mail.channel	email
2	1	1	5	2018-02-17 06:37:34	res.partner	notification
3	1	1	1	2018-02-17 06:38:09	mail.channel	notification
4	1	1	1	2018-02-17 06:38:25	mail.channel	notification
5	1	1	1	2018-02-17 06:39:58	crm.team	notification
6	1	1	1	2018-02-17 06:40:02	mail.channel	notification
7	1	1	6	2018-02-17 06:45:15	res.partner	notification
8	1	1	7	2018-02-17 06:45:56	res.partner	notification
9	1	1	8	2018-02-17 06:46:47	res.partner	notification
10	1	1	9	2018-02-17 06:47:44	res.partner	notification
11	1	1	10	2018-02-17 06:49:06	res.partner	notification
12	1	1	1	2018-02-17 06:52:06	product.template	notification
13	1	1	1	2018-02-17 06:52:06	product.product	notification
14	1	1	2	2018-02-17 07:00:38	product.template	notification
15	1	1	2	2018-02-17 07:00:38	product.product	notification
16	1	1	3	2018-02-17 07:01:03	product.template	notification

**Gambar 4.30** Bentuk isi tabel mail\_message

id	create_date	mail_id	old_value_char	write_field	field_desc	new_value_char	write_date
23	2018-02-17 15:40:	54		1 state	Status	Draft PO	2018-02-17 15
24	2018-02-17 15:40:	54		1 partner_id	Vendor	Amanah Garment	2018-02-17 15
25	2018-02-17 15:40:	55	Draft PO	1 state	Status	Cancelled	2018-02-17 15
26	2018-02-17 15:40:	56	Cancelled	1 state	Status	Draft PO	2018-02-17 15
27	2018-02-17 15:41:	57	Draft PO	1 state	Status	Purchase Order	2018-02-17 15
28	2018-02-17 15:41:	59		1 priority	Priority	Normal	2018-02-17 15
29	2018-02-17 15:41:	59	(Null)	1 date	Creation D	(Null)	2018-02-17 15
30	2018-02-17 15:41:	59		1 state	Status	Draft	2018-02-17 15
31	2018-02-17 15:41:	60	Draft	1 state	Status	Waiting Availabili	2018-02-17 15
32	2018-02-17 15:41:	61	Waiting Availa	1 state	Status	Available	2018-02-17 15
33	2018-02-17 15:42:	62	Available	1 state	Status	Done	2018-02-17 15
34	2018-02-17 15:43:	64		1 state	Status	Draft PO	2018-02-17 15
35	2018-02-17 15:43:	64		1 partner_id	Vendor	Amanah Garment	2018-02-17 15
36	2018-02-17 15:43:	65	Exception	1 state	Status	Running	2018-02-17 15
37	2018-02-17 15:47:	69	(Null)	1 amount_unt	Untaxed A	(Null)	2018-02-17 15

**Gambar 4.31** Bentuk isi tabel mail\_tracking\_value

id	name	company_id	create_date	company_type	street	city	zip	title	c
8	Amanah Garment	1	2018-02-17 06:46:	company	Jl. SD Sekemandu	Bandung	(Null)	(Null)	
9	Vido Garment	1	2018-02-17 06:47:	company	Jl. Pucang Adi No	Surabaya	(Null)	(Null)	
10	Graha Konveksi	1	2018-02-17 06:49:	company	Jl. Ngagel Madya	Surabaya	(Null)	(Null)	
18	Arif	1	2018-02-19 08:21:	person	(Null)	(Null)	(Null)	(Null)	
11	Mahendrawathi	1	2018-02-17 07:18:	person	(Null)	(Null)	(Null)	(Null)	
14	Mila	1	2018-02-18 05:56:	person	(Null)	(Null)	(Null)	(Null)	
3	Administrator	1	2018-02-17 06:36:	person	(Null)	(Null)	(Null)	(Null)	
19	Arif	1	2018-02-19 08:21:	person	(Null)	(Null)	(Null)	(Null)	
12	Susi	1	2018-02-17 15:46:	person	Jl. Kertajaya 100	(Null)	(Null)	(Null)	
15	Mila	1	2018-02-18 05:56:	person	(Null)	(Null)	(Null)	(Null)	
4	Public user	1	2018-02-17 06:36:	person	(Null)	(Null)	(Null)	(Null)	
5	Template User	1	2018-02-17 06:37:	person	(Null)	(Null)	(Null)	(Null)	
1	My Company	1	(Null)	company	(Null)	(Null)	(Null)	(Null)	
6	Greens Productio	1	2018-02-17 06:45:	company	Jl. Perak No.14, C	Bandung	(Null)	(Null)	
7	Pabrik Baju Bandu	1	2018-02-17 06:45:	company	Jl. Andir No.22, D	Bandung	(Null)	(Null)	

**Gambar 4.32** Bentuk isi tabel res\_partner

id	active	login	company_id	partner_id	create_date	create	write	signature	active
1	t	nambisembilu@g	1	3	(Null)	1	1	<span>---<b>(Null)	
5	t	mahendrawathi.e	1	11	2018-02-17 07:18:	1	1	<p> </p>(Null)	
3	f	public	1	4	2018-02-17 06:36:	1	1	(Null)	(Null)
4	f	portaltemplate	1	5	2018-02-17 06:37:	1	1	(Null)	(Null)

**Gambar 4.33** Bentuk isi tabel res\_users

Tabel transaksi yang dilibatkan berhubungan dengan aktivitas dalam proses bisnis. Pada aktivitas “Pembuatan penawaran”. “Konfirmasi penjualan” dan “Selesai penjualan” melibatkan tabel transaksi *sale\_order* untuk mendapatkan informasi *case id*. Untuk aktivitas “Buat tagihan” dan “Bayar tagihan” menggunakan tabel transaksi *account\_invoice*, yang terakhir untuk aktivitas “Persiapan pengiriman” dan “Selesai Pengiriman” memanfaatkan tabel transaksi *stock\_picking* dan *stock\_move*. Selain melibatkan beberapa tabel pengambilan informasi pada studi kasus ini perlu ditambahkan parameter-parameter tertentu agar hasil yang didapatkan tepat dan sesuai.

pg-digitalocean-inspi women\_clothing public Run Stop Explain

```

=trim(res_users.partner_id::text) WHERE mail_tracking_value.write_date IS NOT NULL AND mail_tracking_value.write_date
BETWEEN '2018-01-01' AND '2018-03-31' and account_account.name='Account Receivable' and mail_message.model=
'account.invoice' and mail_tracking_value.new_value_char='Paid' group by case_id,activity,timestmp,originator ) UNION
ALL ( SELECT stock_move.origin as case_id,'Persiapan Pengiriman' as activity,to_char(stock_move.write_date,'YYYY-MM-DD
HH24:MI:SS') as timestmp,res_partner.name as originator FROM mail_tracking_value JOIN mail_message on trim(mail_message
.id::text)=trim(mail_tracking_value.mail_message_id::text) JOIN stock_picking on trim(stock_picking.id::text)=trim(
mail_message.res_id::text) JOIN stock_move on trim(stock_move.picking_id::text)=trim(stock_picking.id::text) JOIN
res_users on trim(res_users.id::text)=trim(mail_message.write_uid::text) JOIN res_partner on trim(res_partner.id::text)
=trim(res_users.partner_id::text) JOIN procurement_rule on trim(procurement_rule.id::text)=trim(stock_move.rule_id::
text) WHERE stock_move.write_date IS NOT NULL AND stock_move.write_date BETWEEN '2018-01-01' AND '2018-03-31' and
mail_message.model='stock.picking' and mail_tracking_value.new_value_char='Draft' and procurement_rule.name='WH: Stock
-> Customers' group by case_id,activity,timestmp,originator ) UNION ALL ( SELECT stock_move.origin as case_id,'Selesai

```

Message Result 1

case_id	activity	timestmp	originator
SO001	Buat Pena	2018-02-17 1	Mahendrawathi
SO001	Konfirmas	2018-02-17 1	Mahendrawathi
SO001	Selesai Pei	2018-02-17 1	Mahendrawathi
SO001	Buat Tagih	2018-02-17 1	Mahendrawathi
SO001	Bayar Tagi	2018-02-17 1	Mahendrawathi
SO001	Persiapan	2018-02-17 1	Mahendrawathi

SELECT \* FROM (SELECT \* FROM ( SELECT sale\_order.name as case\_id,'Buat Penawaran' as activity,to\_char(n Read Only Query time: 0.145s Record 1 of 51

**Gambar 4.34** Contoh hasil ekstraksi *event logs* Odoo ERP dengan SQL

PANEL BAHASA ALAMI

Women Cloting Case ( DBMS : PostgreSQL, DB : women\_clothing, HOST : localhost )

Bahasa Natural :

tampilkan proses order penjualan mulai tanggal 1 januari 2018 sampai tanggal 31 maret 2018

Proses

HASIL EKSTRAKSI BAHASA ALAMI

Waktu ekaekusi program 0.00161 detik

KETERANGAN EKSTRAKSI BAHASA NATURAL

Proses Bisnis : order penjualan  
Aktivitas : buat penawaran, konfirmasi penawaran, buat tagihan, bayar tagihan, persiapan pengiriman, selesai pengiriman, selesai penjualan  
Bulan : Tidak ditemukan.  
Tahun : Tidak ditemukan.  
Tanggal : 1 januari 2018, 31 maret 2018  
Filter Kolom : Tidak ditemukan.  
Urutkan : Tidak ditemukan.

Filter: Type to filter... Show: 10

NO	CASE ID	ACTIVITY/TASK	TIMESTAMP	ORIGINATOR
1	SO001	Buat Penawaran	2018-02-17 15:47:32	Mahendrawathi
2	SO001	Konfirmasi Penawaran	2018-02-17 15:47:38	Mahendrawathi
3	SO001	Selesai Penjualan	2018-02-17 15:47:57	Mahendrawathi
4	SO001	Buat Tagihan	2018-02-17 15:48:21	Mahendrawathi
5	SO001	Bayar Tagihan	2018-02-17 15:48:33	Mahendrawathi
6	SO001	Persiapan Pengiriman	2018-02-17 15:50:09	Mahendrawathi
7	SO001	Selesai Pengiriman	2018-02-17 15:50:09	Mahendrawathi
8	SO002	Buat Penawaran	2018-02-18 05:57:32	Mahendrawathi
9	SO002	Konfirmasi Penawaran	2018-02-18 05:57:43	Mahendrawathi
10	SO002	Selesai Pengiriman	2018-02-18 06:00:55	Mahendrawathi

EXPORT CSV

Showing 1 to 10 of 51 entries

1 2 3 4 5 6

**Gambar 4.35** Contoh hasil ekstraksi *event logs* Odoo ERP dengan NLIDB

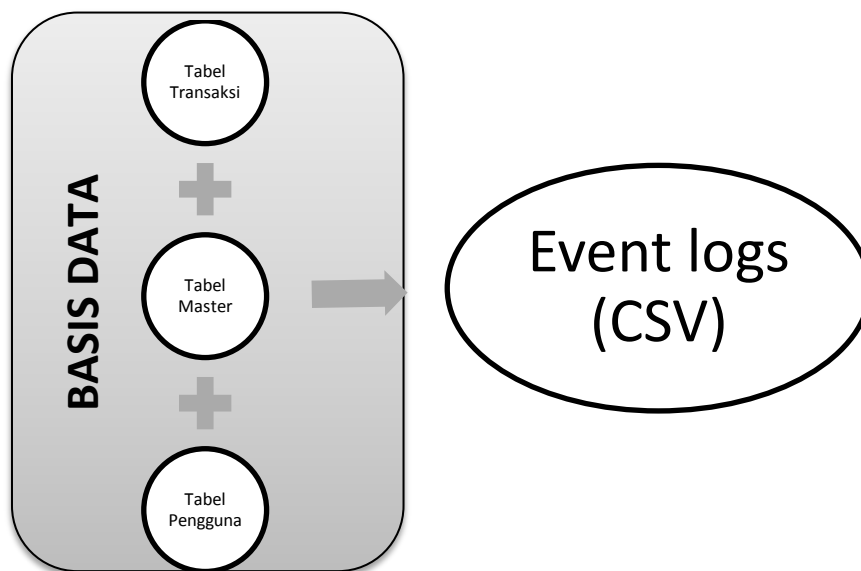
Dari hasil ekstraksi kemudian akan dibandingkan dengan jumlah data awal pada tiap tabel pada basis data sistem Odoo ERP.

**Tabel 4.24** Perbandingan data sistem Odoo ERP

SUMBER DATA	SEBELUM DI EKTRAK		SETELAH DI EKSTRAK		
	TABEL	$\Sigma$	KETERANGAN	$\Sigma$	$\Sigma$
		RECORD		SQL	NLIDB
ODOO ERP	mail_message	307	Mulai tanggal 1 januari 2018 sampai 31 maret 2018	51	51
	mail_tracking_value	366			
	res_partner	20			
	res_users	4			

Pada tabel diatas dapat dilihat jumlah data pada basis data sistem Odoo ERP untuk tabel terkait sebelum dan sesudah dilakukan ekstraksi untuk kebutuhan *event logs*. Dari hasil tersebut dapat menunjukkan bahwa konsep ekstraksi data *event logs* dalam penelitian ini sudah berhasil dijalankan.

Dari beberapa studi kasus yang dilakukan pada penelitian ini didapatkan bahwa untuk mendapatkan sebuah informasi *event log* yang utuh perlu melibatkan beberapa tabel yang memiliki relasi satu sama lain dalam basis data serta tambahan parameter agar hasil yang didapatkan spesifik dengan aktivitas dalam sebuah proses bisnis. Pada penelitian ini menggunakan konsep *query builder* yang diawali dengan proses penentuan beberapa tabel terkait disertai dengan relasi masing-masing tabel kemudian ditambahkan parameter jika dibutuhkan. Dari pemetaan tersebut akan menghasilkan sebuah bahasa SQL yang kemudian digunakan untuk ekstraksi informasi *event log* tiap aktivitas. Gambaran proses ekstraksi dapat dilihat pada gambar 4.34, konsep ini memungkinkan pengguna sistem evenl untuk melakukan ekstraksi event log secara realtime dalam sebuah sistem yang berjalan. Bentuk hasil ekstraksi *event logs* yang dihasilkan pada penelitian ini dapat dilihat lempiran F. Bentuk hasil ekstraksi tersebut berbentuk tabel dengan kolom atribut *event logs* yang dapat diekspor kedalam format CSV.



**Gambar 4.36** Gambaran proses ekstraksi dan konversi event logs dari basis data

Pada pengujian bagian 4.6.1 dengan menguji ekstraksi *event logs* pada dua sistem dengan beberapa bentuk KPE dan menggunakan metode pengujian *black box* memberikan hasil ekstraksi menggunakan bahasa alami yang sesuai ditunjukkan dengan gambar 4.9 sampai dengan gambar 4.14. Antarmuka yang digunakan dalam penelitian ini yaitu antarmuka bahasa alami yang terstruktur atau dikenal NLIDB (*Natural Language Interface to a Database*). Bahasa alami yang digunakan pada penelitian ini juga dapat mengakomodir kebutuhan konversi atribut waktu untuk penyaringan data *event log*. Atribut waktu dalam bahasa alami yang digunakan yaitu tanggal dengan format bahasa indonesia, periode waktu dengan batasan dua tanggal, serta bulan pada tahun tertentu. Proses konversi atribut waktu pada bahasa alami dalam penelitian ini memanfaatkan kata bantu keterangan waktu untuk mendapatkan nilai atribut waktu yang dimaksud. Pada beberapa pengujian untuk atribut waktu bahasa alami dalam penelitian ini sudah dapat menunjukkan keterangan *error* jika format tanggal tidak sesuai ataupun terdapat kesalahan penulisan format tanggal.

Selanjutnya pada pengujian pada 4.6.2 konversi format CSV ke dalam format XES dengan bantuan web <http://www.satcos.in/processmining/index.php> didapatkan bahwa hasil CSV pada penelitian ini dapat dikonversi ke dalam format XES dengan baik. Namun ketika hasil tersebut di masukan ke dalam aplikasi ProM didapatkan error dalam pembacaan format tersebut, hal ini dikarena perbedaan

pendefinisian format XES pada aplikasi ProM dengan XES yang dihasilkan dalam web *satcos*. Di lain sisi pada ketika format CSV yang dihasilkan pada penelitian ini langsung dimasukan ke dalam aplikasi ProM import tidak mendapatkan error dalam arti berhasil digunakan untuk keperluan *process mining*. Pada pengujian lain dengan menggunakan aplikasi *disco* hasil CSV yang didapatkan penelitian ini tidak mengalami kendala error baik saat dikonversi ke dalam format XES maupun ketika digunakan aplikasi ProM.

Pada pengujian nilai akurasi aplikasi dalam penelitian ini mendapatkan nilai akurasi 84% hasil tersebut tergolong baik dan menunjukkan aplikasi ekstraksi event log yang dihasilkan dalam penelitian ini layak untuk digunakan. Kesalahan yang terjadi saat proses ekstraksi event logs karena banyak kesalahan penulisan proses bisnis dan aktivitas serta ketidak sesuaian pada penulisan struktur bahasa alami. Oleh karena itu dengan tambahan fitur otomatis saran dan otomatis perbaikan pada antarmuka bahasa alami semakin menambahkan kemudahan dalam proses ekstraksi dengan menggunakan bahasa alami. Pada fitur perbaikan otomatis pada frasa proses bisnis dan aktivitas dalam penelitian ini menguji kinerja dua algoritma (Oliver, 1994) atau dikenal dengan fungsi *similar\_text* pada bahasa pemrograman PHP dan algoritma *Longest Common Subsequences* (LCS) oleh (Paterson dan Dančik, 1994). Dari perbandingan dua algoritma tersebut didapatkan bahwa algoritma LCS memiliki kinerja yang lebih baik dalam hal perbaikan frasa aktivitas proses bisnis dan aktivitas. Kemudian untuk menunjukkan hasil validasi aplikasi ekstraksi event logs dalam penelitian ini dilakukan proses UAT dengan pakar analis bisnis dari 4 poin yang diujikan didapatkan aplikasi ini sudah sesuai dengan kebutuhan analis bisnis.

Pada penelitian ini berhasil membuat sebuah aplikasi untuk ekstraksi yang tersimpan dalam basis data dengan menggunakan antarmuka bahasa alami yang terstruktur. Metode untuk proses ekstraksi menggunakan NLIDB. Dari dua studi kasus yang sudah pada proses bisnis yang sudah ditentukan sebelumnya, hasil event log yang didapatkan dalam penelitian sudah bisa digunakan untuk kebutuhan *process mining* oleh analis bisnis. Hasil yang didapatkan dalam penelitian ini melanjutkan penelitian penggunaan NLIDB untuk pengambilan informasi dari basis data, penelitian sebelumnya yang dilakukan oleh (Reinaldha

dan Widagdo, 2014) yang menangani konversi satuan dan menangani kalimat Bahasa Indonesia berbentuk pertanyaan. Untuk penelitian ini berhasil memberikan kontribusi penggunaan bahasa alami yang mengandung unsur atribut waktu, dengan memanfaatkan susunan kata dalam sebuah kalimat di dalam Bahasa Indonesia yang diikuti kata keterangan waktu sehingga dapat mengidentifikasi nilai atribut waktu yang ada dalam bahasa alami. Beberapa unsur atribut waktu yang berhasil di baca dalam penelitian ini yaitu atribut tanggal, periode berdasarkan tanggal, bulan dan tahun. Proses ekstraksi bahasa alami yang mengandung unsur diharapkan dapat memberikan gambaran untuk penelitian lain di bidang NLIDB.

Kontribusi lain dalam penelitian ini yaitu aplikasi praktis yang digunakan oleh analis bisnis ataupun pihak lain yang membutuhkan ekstraksi event log yang tersimpan dalam basis data. Penelitian ini melanjutkan penelitian yang sebelumnya dilakukan oleh (J. Buijs, 2010; Jans dkk., 2012; Piessens, 2011) dalam pembuatan aplikasi untuk melakukan ekstraksi event log dalam sebuah basis data. Kelanjutan pada penelitian ini dibandingkan penelitian sebelumnya yaitu mengurangi proses analisa informasi *event logs* yang tersimpan di basis data dilakukan berkali kali menjadi dilakukan hanya sekali saja selain itu antarmuka yang digunakan adalah bahasa alami yang lebih dipahami oleh analis bisnis. Kemudian hasil dari penelitian ini juga dapat digunakan untuk sistem pada umumnya tidak hanya digunakan untuk sistem SAP saja. Hal tersebut dibuktikan dengan sistem yang berhasil diuji dalam penelitian ini yaitu SIM Rumah Sakit dan Odoo ERP. Perbandingan hasil penelitian ini dibandingkan penelitian sebelumnya dalam ekstraksi dan konversi *event logs* untuk kebutuhan *process mining* dapat dilihat di tabel dibawah ini



**Tabel 4.26** Tabel perbandingan penelitian ekstraksi *event logs*

Pengarang	Proses	Antarmuka	Scope	Keakuratan
(Bujis, 2010)	Ekstraksi event logs dalam basis data dilakukan berulang	form	SAP	Tidak diuji
(Peessen, 2011)	Ekstraksi event logs dalam basis data dilakukan berulang	form	SAP	Tidak diuji
Penelitian ini	Ekstraksi event logs dalam basis data dilakukan berulang	Bahasa alami	SISTEM RUMAH SAKIT, OPEN SOURCE ERP	0.844

Proses ekstraksi yang dilakukan dalam penelitian ini yaitu dengan memetakan informasi *event logs* dalam basis data dari sebuah sistem kemudian memasukan perintah bahasa alami dengan Bahasa Indonesia selanjutnya hasil event logs akan ditampilkan dalam bentuk tabulasi yang bisa di ekspor ke dalam format CSV. Dari hasil beberapa pengujian yang sudah dilakukan ini aplikasi ini siap digunakan untuk ekstraksi event log. Untuk mengetahui gambaran hasil dari awal proses ekstraksi *event logs* sampai dapat dipakai dalam kebutuhan *process mining* pada penelitian ini dapat dilihat di lampiran E.



*Halaman ini sengaja dikosongkan*

## **BAB V**

### **KESIMPULAN**

Dalam bab ini akan kesimpulan dari hasil uji coba dan analisis hasil yang dilakukan pada tahapan scenario pengujian

#### **5.1 Kesimpulan**

Setelah penelitian ekstrak event logs dengan menggunakan bahasa alami pada sistem informasi yang telah selesai dilakukan, maka berikut ini kesimpulan dari penelitian ini.

1. Untuk melakukan ekstraksi *event logs* dalam basis data sebuah sistem penelitian ini menggunakan bahasa alami sebagai masukkannya. Kalimat perintah bahasa alami yang digunakan dibagi menjadi beberapa jenis yang disebut dalam penelitian ini KPE (Kalimat Perintah Ekstraksi). Perintah tersebut diawali dengan kata perintah yaitu “tampilkan” diikuti dengan frasa proses bisnis atau aktivitas yang telah ditentukan sebelumnya. Metode yang digunakan untuk melakukan proses tersebut yaitu NLIDB. Dengan menggunakan metode tersebut bahasa alami tersebut akan dirubah kedalam bahasa SQL kemudian akan dibaca oleh DBMS sehingga dapat mengembalikan sebuah data *event logs* dengan atribut (caseid, activity, timestamp dan originator). Data tersebut diproses kembali menggunakan bahasa pemrograman PHP agar dapat di ekspor menjadi format CSV. Selanjutnya untuk mengkonversi format CSV menjadi XES (standar format *event logs*) dilakukan dengan program *Disco* ataupun aplikasi berbasis web dengan alamat <http://www.satcos.in/processmining/index.php>.
2. Proses ekstraksi berbasis bahasa alami masih memiliki kelemahan yaitu kesalahan penulisan struktur kalimat dan kesalahan penulisan ejaan kata atau frasa. Pada penelitian ini menggunakan fungsi otomatis saran dan otomatis perbaikan pada antarmuka masukan bahasa alami dalam rangka menutupi kelemahan tersebut. Pada proses perbaikan frasa penelitian ini membandingkan kinerja algoritma *similar text* (Oliver,

1994) dan LCS (Paterson dan Dančik, 1994), dari perbandingan tersebut didapatkan algoritma LCS memberikan hasil yang lebih baik dengan nilai rata-rata kurva ROC sebesar 0.910 atau 91 %. Hal ini membantu menutupi kekurangan tersebut selain itu dapat memudahkan pengguna dalam penggunaan aplikasi untuk kebutuhan ekstraksi *event log*.

3. Untuk mengakomodir konversi atribut waktu dalam bahasa alami pada penelitian ini membuat daftar format waktu (tanggal, bulan, tahun dan periode) sesuai dengan kaidah bahasa Indonesia, kemudian masukkan yang bahasa alami yang sesuai daftar tersebut akan dikonversi dalam format waktu dalam komputer dengan bantuan fungsi (*date* dan *strtotime*) pada bahasa pemrograman PHP. Konversi tersebut dilakukan karena format waktu yang ditulis dengan bahasa alami hanya sebatas tanggal bulan dan tahun akan tetapi tipe data yang digunakan dalam basis data pada penelitian ini menggunakan *timestamp* (terdapat unsur jam, menit dan detik).
4. Dengan menggunakan bahasa alami yang terstruktur pada penelitian penelitian ini berhasil melakukan ekstraksi *event logs* yang tersimpan dalam basis data Odoo ERP dan SIM Rumah Sakit dengan nilai akurasi sebesar 0.844 atau 84 %. Kemudian ditunjukkan dengan grafis kurva ROC dengan nilai sensitivitas optimum yang bernilai 0.844 atau 84 % serta nilai spesifitas optimum 0.436 atau 44 %. Dari kurva memperlihatkan kinerja ekstraksi *event logs* dengan bahasa alami diatas garis *baseline* sehingga dapat dikatakan memiliki kinerja yang baik. Selain itu hasil ekstraksi dapat digunakan secara langsung tanpa harus melakukan analisa informasi event log dalam basis yang sedang berjalan dalam sebuah sistem secara berkali-kali. Kemudian hasil ekstraksi yang bentuk tabel dapat diekspor ke dalam format CSV dan bisa dimasukkan kedalam aplikasi ProM untuk kebutuhan *process mining*.

## 5.2 Saran

Penelitian ini memiliki keterbatasan dalam proses pemetaan yang dilakukan semi otomatis atau dalam arti dengan melibatkan seorang analis sistem yang mengerti struktur basis data dalam sebuah sistem tersebut. Sehingga ketika tanpa bantuan pengetahuan oleh sistem analis aplikasi ini tidak bisa berjalan dengan baik. Hal tersebut dilakukan karena struktur penamaan tabel yang ada dalam sistem berbeda-beda dan tidak ada standar dalam penamaan tabel dalam sebuah basis data. Oleh karena itu pada penelitian selanjutnya diharapkan bisa menutupi kekurangan pada penelitian ini yaitu dengan membuat sebuah pemetaan otomatis dengan menggunakan metode *machine learning* maupun metode lain yang lebih handal.

*Halaman ini sengaja dikosongkan*





## DAFTAR PUSTAKA

- Aalst, W. M. P. van der, (2012): Process mining manifesto, *7th International Workshop on Business Process Intelligence (BPI 2011)*, 169–194, Campus des Cézeaux, Clermont-Ferrand: Springer-Verlag, diperoleh melalui situs internet: <http://eprints.qut.edu.au/47515/>.
- Aguilar-Saven, R. S. (2004): Business process modelling: Review and framework, *International Journal of production economics*, **90**(2), 129–149.
- Andayani, S. (2002): *Query Bahasa Indonesia untuk Basis Data Akademik*, Tesis Magister Komputer, Program Pasca Sarjana Ilmu Komputer, UGM, Yogyakarta.
- Backus, J. W., Bauer, F. L., Green, J., Katz, C., McCarthy, J., Naur, P., Perlis, A. J., Rutishauser, H., Samelson, K., Vauquois, B., dan others (1963): Revised report on the algorithmic language Algol 60, *The Computer Journal*, **5**(4), 349–367.
- Betz, S., Hickl, S., dan Oberweis, A. (2011): Risk-aware business process modeling and simulation using XML nets, *Commerce and enterprise computing (cec)*, *2011 IEEE 13th conference on*, 349–356, IEEE.
- Bose, R. J. C., Mans, R. S., dan van der Aalst, W. M. (2013): Wanna improve process mining results?, *Computational Intelligence and Data Mining (CIDM)*, *2013 IEEE Symposium on*, 127–134, IEEE.
- Buijs, i J. C. A. M. (2010): Master's Thesis, *Science*.
- Buijs, J. (2010): *Mapping Data Sources to XES in a Generic Way*, Master's thesis, Eindhoven University of Technology.
- Calvanese, D., Montali, M., Syamsiyah, A., dan van der Aalst, W. M. P. (2015): Ontology-Driven Extraction of Event Logs from Relational Databases, *Proc. of the 11th Int. Workshop on Business Process Intelligence (BPI 2015)*, Springer.
- de Murillas, E. G. L., van der Aalst, W. M., dan Reijers, H. A. (2015): Process Mining on Databases: Unearthing Historical Data from Redo Logs, *Business Process Management*, 367–385, Springer.

- Greg Alexander, Aaron Bartels, dan Mike Drapeau (2008): *Making the Number: How to Use Sales Benchmarking to Drive Performance*, Penguin Group.
- Günther, C. W., dan van der Aalst, W. M. (2006): A generic import framework for process event logs, *Business Process Management Workshops*, 81–92, Springer.
- Guo, J., Foo, K., Barbour, L., dan Zou, Y. (2008): A business process explorer, *Software Engineering, 2008. ICSE'08. ACM/IEEE 30th International Conference on*, 871–874, IEEE.
- Hartati, S., dan Zuliarso, E. (2011): Aplikasi Pengolah Bahasa Alami untuk Query Basisdata XML.
- Ingvaldsen, J. E. (2011): ontology-Driven Business Process intelligence, *Applied Semantic Web Technologies*, 395.
- Ingvaldsen, J. E., dan Gulla, J. A. (2007): Preprocessing support for large scale process mining of SAP transactions, *Business Process Management Workshops*, 30–41, Springer.
- Jans, M., Alles, M., dan Vasarhelyi, M. (2012): Process mining of event logs in internal auditing: a case study.
- Jeston, J., dan Nelis, J. (2014): *Business process management*, Routledge.
- Jung, J.-Y., Bae, J., dan Liu, L. (2008): Hierarchical business process clustering, *Services Computing, 2008. SCC'08. IEEE International Conference on*, **2**, 613–616, IEEE.
- Kennedy, R. E., dan Sharma, A. (2009): *The services shift: seizing the ultimate offshore opportunity*, FT Press.
- Khodabandelou, G., Hug, C., Deneckere, R., dan Salinesi, C. (2013): Process mining versus intention mining, *Enterprise, Business-Process and Information Systems Modeling*, 466–480, Springer.
- Laguna, M., dan Marklund, J. (2013): *Business process modeling, simulation and design*, CRC Press.
- Mardhatillah, L., Er, M., dan Kusumawardani, R. P. (2012): Identifikasi Bottleneck pada Hasil Ekstraksi Proses Bisnis ERP dengan Membandingkan Algoritma Alpha++ dan Heuristics Miner, *Jurnal Teknik ITS*, **1**(1), A322–A327.

- Marquez, G., Rodriguez, A., dan Medina, E. F. (2014): Obtaining secure BPEL from Secure Business Process specified with BPMN, *IEEE Latin America Transactions*, **12**(2), 315–320.
- Medeiros, A. K. A. (2004): *Process Mining: Extending the [alpha]-algorithm to Mine Short Loops*, Beta, Research School for Operations Management and Logistics, diperoleh melalui situs internet: <https://books.google.co.id/books?id=ZqA8MwEACAAJ>.
- Nakatumba, J., dan van der Aalst, W. M. (2009): Analyzing resource behavior using process mining, *International Conference on Business Process Management*, 69–80, Springer.
- Oliver, I. (1994): Programming classics: implementing the world's best algorithms.
- Paterson, M., dan Dančik, V. (1994): Longest common subsequences, *International Symposium on Mathematical Foundations of Computer Science*, 127–142, Springer.
- Piessens, D. (2011): Event log extraction from SAP ECC 6.0.
- Poonia, V. (2010): *Production and Operation Management*, Gennext Publications.
- Raharjo, S., dan Hartati, S. (2014): ANTARMUKA BAHASA ALAMI UNTUK MELAKUKAN QUERY TERHADAP TERJEMAHAN AL-QURAN, *Jurnal Teknologi*, **7**.
- Reinaldha, F., dan Widagdo, T. E. (2014): Natural Language Interfaces to Database (NLIDB): Question handling and unit conversion, *Data and Software Engineering (ICODSE), 2014 International Conference on*, 1–6, IEEE.
- Saylam, R., dan Sahingoz, O. K. (2013): Process mining in business process management: Concepts and challenges, *Electronics, Computer and Computation (ICECCO), 2013 International Conference on*, 131–134, IEEE.
- Scheer, I. (2006): Aris platform, *IDS Scheer website*, <http://www.idsscheer.com/us/products/aris-platform.htm> (accessed 13.11. 2006).

- van der Aalst, W. M. (2015): Extracting event data from databases to unleash process mining, *BPM-Driving innovation in a digital world*, 105–128, Springer.
- van der Aalst, W. M. P. (2011): *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, 1st ed., Springer Publishing Company, Incorporated.
- Van Der Aalst, W., dan Van Hee, K. M. (2004): *Workflow management: models, methods, and systems*, MIT press.
- Walicki, M., dan Ferreira, D. R. (2011): Sequence partitioning for process mining with unlabeled event logs, *Data & Knowledge Engineering*, **70**(10), 821–841.
- Wang, D., Ge, J., Hu, H., dan Luo, B. (2011): A new process mining algorithm based on event type, *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*, 1144–1151, IEEE.
- Wang, D., Ge, J., Hu, H., Luo, B., dan Huang, L. (2012): Discovering Process Models from Event Multiset, *Expert Syst. Appl.*, **39**(15), 11970–11978, diperoleh melalui situs internet: <https://doi.org/10.1016/j.eswa.2012.03.064>.
- Wibisono, S. (2013): Aplikasi pengolah bahasa alami untuk query basisdata akademik dengan format data xml, *Dinamik-Jurnal Teknologi Informasi*, **18**(1).

*Halaman ini sengaja dikosongkan*



## LAMPIRAN A

### Bukti ijin pengambilan data untuk penelitian



## UNIVERSITAS AIRLANGGA RUMAH SAKIT

Kampus C Unair Surabaya 60115 Telp. 031-5916290, 031-58208281, 031- 81153153  
Fax. 031-5916291 Website : rumahsakit.unair.ac.id, email : sekretariat@rsua.unair.ac.id

3 Agustus 2017

No. : 1628 /UN3.9.1/PPd/2017  
Hal. : Jawaban Permohonan Ijin Survey dan Pengambilan Data  
an Nambi Sembilu, S.Kom

Kepada Yth.  
Ketua Departemen Sistem Informasi  
Fakultas Teknologi Informasi ITS Surabaya

Menjawab surat tanggal 15 Juni 2017 dengan nomor 035638/IT2.VI.7.2/PM.05/2017 tentang permohonan survey dan pengambilan data di RS Universitas Airlangga an:

Nama : **Nambi Sembilu, S.Kom**  
NRP : 5214201004

Judul Penelitian : Ekstraksi dan Konversi *Event Log* pada *Temporal Database* di dalam Sebuah Sistem Informasi dengan Menggunakan Bahasa Alami yang Terstruktur

Dengan ini disampaikan bahwa :

- 1) Permohonan ijin diberikan karena proposal penelitiannya telah dinyatakan Laik Etik oleh Komite Etika dan Hukum RSUA,
- 2) Ijin penelitian digunakan untuk kepentingan penulisan tesis dan tidak akan dipublikasikan tanpa ijin tertulis dari Direktur RSUA,
- 3) Untuk pengambilan data alur pelayanan di farmasi mohon berkoordinasi dengan Manajer Farmasi RSUA yaitu Dr.Budi Suprpti, Dra., M.Si
- 4) Untuk pengambilan data base sistem informasi RSUA mohon berkoordinasi dengan Ahmad Fuad Halimi, ST
- 5) Menandatangani Perjanjian Pernyataan Kerahasiaan (bermaterai),
- 6) Selesai penelitian, wajib menyerahkan Laporan ke RSUA

Demikian yang dapat saya sampaikan. Atas perhatian Saudara, Saya sampaikan terima kasih

Wakil Direktur Pendidikan & Riset,

Prof. Dr.Muhammad Amin,dr.,Sp.P (K)  
NIP. 194708101974121002

Tembusan : Yth

1. Ketua Komite Etika & Hukum RSUA
2. Manajer Farmasi RSUA
3. Ahmad Fuad Halimi, ST
- 4. **Nambi Sembilu, S.Kom**

File : SK KEH RSUA



**KOMITE ETIKA DAN HUKUM  
RUMAH SAKIT UNIVERSITAS AIRLANGGA**

KETERANGAN LOLOS KAJI ETIK  
No :121/KEH/2017

Komite Etika dan Hukum Rumah Sakit Universitas Airlangga dalam upaya melindungi hak asasi dan kesejahteraan subyek penelitian, telah mengkaji dengan teliti proposal penelitian dan presentasi proposal yang berjudul :

Ekstraksi dan Konversi *Event Log* pada *Temporal Database* di dalam Sebuah Sistem Informasi dengan Menggunakan Bahasa Alami yang Terstruktur

Peneliti : **Nambi Sembilu, S.Kom**

Institusi Penelitian : Rumah Sakit Universitas Airlangga

DINYATAKAN LAIK ETIK.

Surabaya 3 Agustus 2017

Ketua

Prof. Dr.Med. H.M.Soekry Erfan Kusuma, dr.Sp.F (K), DFM



*Halaman ini sengaja dikosongkan*



## LAMPIRAN B

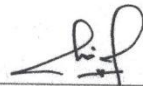
Pengujian UAT dengan analisis bisnis

Tesis Ekstraksi Eventlog Menggunakan Bahasa Alami						Requirement Validation			
Institut Teknologi Sepuluh Nopember Prodi Magister Sistem Informasi									
Nama Proyek	Event Web		Di Review Oleh:		1. Dr.Eng. Febrillyan Samopa, S.Kom., M.Kom. 2. Mahendrawathi ER, ST., MSc., PhD.				
Versi	1.0		Tgl. Tes:		15 Maret 2018				
Lingkungan Tes	Odoo ERP		Disetujui Oleh:		Mahendrawathi ER, ST., MSc., PhD.				
Jenis Tes	Requirement Validation		Tgl. Persetujuan		15 Maret 2018				
Dibuat Oleh	Nambi Sembilu (5214201004)								
Dibuat Tanggal	13 Februari 2018								

MODUL	NOMOR SKENARIO	NOMOR KASUS	LANGKAH TES	DATA YANG DIGUNAKAN	HASIL PROSEDUR YANG DIHARAPKAN	HASIL AKTUAL	BERHASIL GAGAL (B/G)	TGL PELAKSANA AAN	PELAKSANA
Manajemen Penjualan	1	1.1	Ekstraksi Proses Bisnis	Order Penjualan	Menampilkan data event log pada proses bisnis order penjualan	Menampilkan Sales Quote	B	15/03/2018	Nambi
		1.2	Ekstraksi Proses Bisnis dengan tambahan pengurutan atribut event log	Order Penjualan	Menampilkan data event log pada proses bisnis dengan tambahan pengurutan atribut waktu	Menampilkan Sales Quote & sales orderurut berdasarkan timestamp	B	15/03/2018	Nambi
		1.3	Ekstraksi beberapa Aktivitas dalam satu proses bisnis	Order Penjualan	Menampilkan data event log berdasarkan beberapa aktivitas pada proses bisnis order penjualan	Menampilkan sales quote, sales order dan invoice	B	15/03/2018	Nambi
		1.4	Ekstraksi Proses Bisnis / Aktivitas dengan filter atribut waktu	Order Penjualan	Menampilkan data event log berdasarkan beberapa aktivitas/ proses bisnis dengan batasan atribut waktu	Menampilkan Sales Quote tanggal 17 - 18 Februari 2018	B	15/03/2018	Nambi

PIC signature:



Date:

15 Maret 2018

Supervisor signature:

Date:

*Halaman ini sengaja dikosongkan*



## LAMPIRAN C

### ROC Curve – Ekstraksi *event logs* dengan bahasa alami

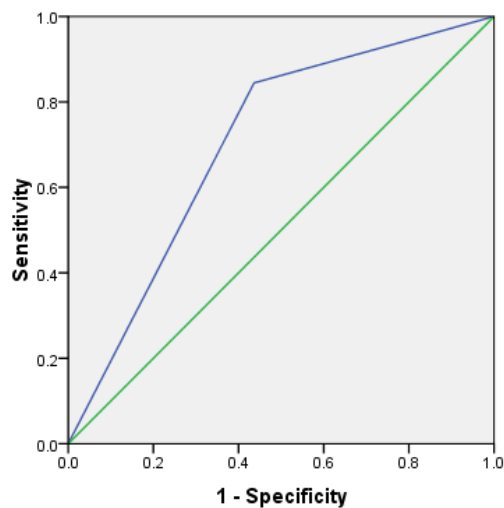
Case Processing Summary

HASIL	Valid N (listwise)
Positive <sup>a</sup>	45
Negative	55

Larger values of the test result variable(s) indicate stronger evidence for a positive actual state.

a. The positive actual state is 1.00.

ROC Curve



Diagonal segments are produced by ties.

Area Under the Curve

Test Result Variable(s): PREDIKSI

Area	Std. Error <sup>a</sup>	Asymptotic Sig. <sup>b</sup>	Asymptotic 95% Confidence Interval	
			Lower Bound	Upper Bound
.704	.053	.000	.601	.807

The test result variable(s): PREDIKSI has at least one tie between the positive actual state group and the negative actual state group. Statistics may be biased.

a. Under the nonparametric assumption

b. Null hypothesis: true area = 0.5

### Coordinates of the Curve

Test Result Variable(s): PREDIKSI

Positive if Greater Than or Equal To <sup>a</sup>	Sensitivity	1 - Specificity
-1.0000	1.000	1.000
.5000	.844	.436
2.0000	.000	.000

The test result variable(s): PREDIKSI has at least one tie between the positive actual state group and the negative actual state group.

a. The smallest cutoff value is the minimum observed test value minus 1, and the largest cutoff value is the maximum observed test value plus 1. All the other cutoff values are the averages of two consecutive ordered observed test values.



*Halaman ini sengaja dikosongkan*



## LAMPIRAN D

### ROC Curve – Algoritma Oliver (PHP similar\_text)

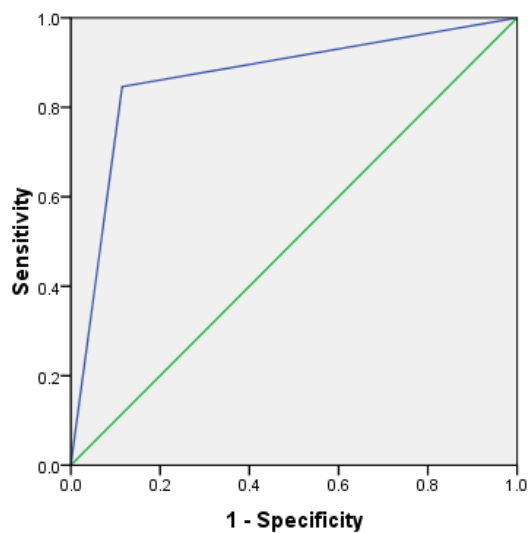
**Case Processing Summary**

HASIL_M1	Valid N (listwise)
Positive <sup>a</sup>	13
Negative	87

Larger values of the test result variable(s) indicate stronger evidence for a positive actual state.

a. The positive actual state is 1.00.

**ROC Curve**



Diagonal segments are produced by ties.

**Area Under the Curve**

Test Result Variable(s):Prediksi

Area	Std. Error <sup>a</sup>	Asymptotic Sig. <sup>b</sup>	Asymptotic 95% Confidence Interval	
			Lower Bound	Upper Bound
.866	.061	.000	.745	.986

The test result variable(s): Prediksi has at least one tie between the positive actual state group and the negative actual state group. Statistics may be biased.

a. Under the nonparametric assumption

### Area Under the Curve

Test Result Variable(s):Prediksi

Area	Std. Error <sup>a</sup>	Asymptotic Sig. <sup>b</sup>	Asymptotic 95% Confidence Interval	
			Lower Bound	Upper Bound
.866	.061	.000	.745	.986

The test result variable(s): Prediksi has at least one tie between the positive actual state group and the negative actual state group. Statistics may be biased.

b. Null hypothesis: true area = 0.5

### Coordinates of the Curve

Test Result Variable(s):Prediksi

Positive if Greater Than or Equal To <sup>a</sup>	Sensitivity	1 - Specificity
-1.0000	1.000	1.000
.5000	.846	.115
2.0000	.000	.000

The test result variable(s): Prediksi has at least one tie between the positive actual state group and the negative actual state group.

a. The smallest cutoff value is the minimum observed test value minus 1, and the largest cutoff value is the maximum observed test value plus 1. All the other cutoff values are the averages of two consecutive ordered observed test values.

## ROC Curve - Algoritma *Longest Common Subsequences*

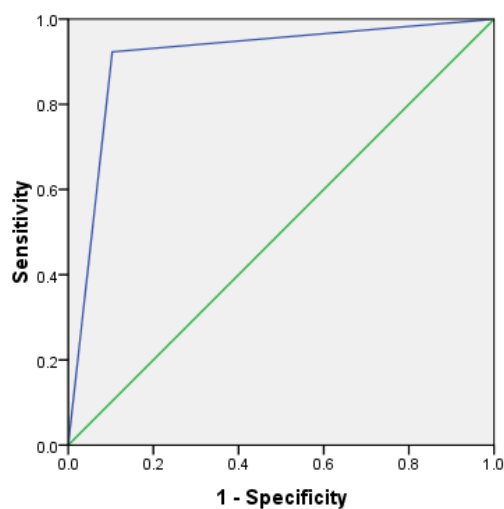
**Case Processing Summary**

HASIL_M2	Valid N (listwise)
Positive <sup>a</sup>	13
Negative	87

Larger values of the test result variable(s) indicate stronger evidence for a positive actual state.

a. The positive actual state is 1.00.

**ROC Curve**



Diagonal segments are produced by ties.

**Area Under the Curve**

Test Result Variable(s):Prediksi

Area	Std. Error <sup>a</sup>	Asymptotic Sig. <sup>b</sup>	Asymptotic 95% Confidence Interval	
			Lower Bound	Upper Bound
.910	.047	.000	.818	1.002

The test result variable(s): Prediksi has at least one tie between the positive actual state group and the negative actual state group. Statistics may be biased.

a. Under the nonparametric assumption

b. Null hypothesis: true area = 0.5

### Coordinates of the Curve

Test Result Variable(s):Prediksi

Positive if Greater Than or Equal To <sup>a</sup>	Sensitivity	1 - Specificity
-1.0000	1.000	1.000
.5000	.923	.103
2.0000	.000	.000

The test result variable(s): Prediksi has at least one tie between the positive actual state group and the negative actual state group.

a. The smallest cutoff value is the minimum observed test value minus 1, and the largest cutoff value is the maximum observed test value plus 1. All the other cutoff values are the averages of two consecutive ordered observed test values.

*Halaman ini sengaja dikosongkan*





## LAMPIRAN E

Contoh bentuk Hasil ekstraksi event logs berbentuk tabel untuk sistem Odoo ERP.

case_id	activity	timestmp	originator
SO001	Buat Penawaran	2/17/2018 15:47	Mahendrawathi
SO001	Konfirmasi Penawaran	2/17/2018 15:47	Mahendrawathi
SO001	Selesai Penjualan	2/17/2018 15:47	Mahendrawathi
SO001	Buat Tagihan	2/17/2018 15:48	Mahendrawathi
SO001	Bayar Tagihan	2/17/2018 15:48	Mahendrawathi
SO001	Persiapan Pengiriman	2/17/2018 15:50	Mahendrawathi
SO001	Selesai Pengiriman	2/17/2018 15:50	Mahendrawathi
SO002	Buat Penawaran	2/18/2018 5:57	Mahendrawathi
SO002	Konfirmasi Penawaran	2/18/2018 5:57	Mahendrawathi
SO002	Selesai Pengiriman	2/18/2018 6:00	Mahendrawathi
SO002	Persiapan Pengiriman	2/18/2018 6:00	Mahendrawathi
SO003	Buat Penawaran	2/18/2018 8:12	Mahendrawathi
SO003	Konfirmasi Penawaran	2/18/2018 8:13	Mahendrawathi
SO003	Selesai Pengiriman	2/18/2018 8:15	Mahendrawathi
SO003	Persiapan Pengiriman	2/18/2018 8:15	Mahendrawathi
SO004	Buat Penawaran	2/18/2018 8:16	Mahendrawathi
SO004	Konfirmasi Penawaran	2/18/2018 8:16	Mahendrawathi
SO004	Selesai Pengiriman	2/18/2018 8:16	Mahendrawathi
SO004	Persiapan Pengiriman	2/18/2018 8:16	Mahendrawathi
SO002	Buat Tagihan	2/18/2018 8:20	Mahendrawathi
SO002	Bayar Tagihan	2/18/2018 8:20	Mahendrawathi
SO002	Persiapan Pengiriman	2/18/2018 8:22	Mahendrawathi
SO002	Selesai Pengiriman	2/18/2018 8:22	Mahendrawathi
SO005	Buat Penawaran	2/18/2018 13:49	Mahendrawathi
SO005	Konfirmasi Penawaran	2/18/2018 13:49	Mahendrawathi
SO005	Selesai Pengiriman	2/18/2018 13:49	Mahendrawathi
SO005	Persiapan Pengiriman	2/18/2018 13:49	Mahendrawathi
SO006	Buat Penawaran	2/18/2018 14:28	Administrator
SO007	Buat Penawaran	2/19/2018 8:33	Mahendrawathi
SO007	Konfirmasi Penawaran	2/19/2018 8:33	Mahendrawathi
SO007	Persiapan Pengiriman	2/19/2018 8:34	Mahendrawathi
SO007	Selesai Pengiriman	2/19/2018 8:34	Mahendrawathi
SO007	Buat Tagihan	2/19/2018 8:38	Mahendrawathi
SO007	Bayar Tagihan	2/19/2018 8:38	Mahendrawathi
SO005	Buat Tagihan	2/19/2018 8:39	Mahendrawathi
SO005	Bayar Tagihan	2/19/2018 8:39	Mahendrawathi
.....	.....	.....	.....

Contoh hasil ekstraksi event logs yang berasal dari basis data sistem Rumah Sakit

case_id	activity	timestmp	originator
7446	Penerimaan Barang	5/1/2017 4:45	Anisa Dwi Syafitri
7446	Persetujuan Permintaan	5/1/2017 4:45	Muhammad Fuad Yasin
7455	Permintaan Barang	5/1/2017 8:24	Nurul Kamariyah S
7456	Permintaan Barang	5/1/2017 8:44	Muhammad Syamsul Hidayat
7457	Permintaan Barang	5/1/2017 11:03	MARIA ULFA
7458	Permintaan Barang	5/1/2017 13:04	Ratna Nisaâ€™ Anggraini
7459	Permintaan Barang	5/1/2017 13:41	Farida Fuji Astutik
7460	Permintaan Barang	5/1/2017 18:13	Ratna Nisaâ€™ Anggraini
7461	Permintaan Barang	5/1/2017 18:21	Ratna Nisaâ€™ Anggraini
7462	Permintaan Barang	5/2/2017 8:16	Anisa Adyalina
7463	Permintaan Barang	5/2/2017 8:54	Agus Arianto
7463	Persetujuan Permintaan	5/2/2017 8:58	Okki Fajrin Dhisati
7463	Penerimaan Barang	5/2/2017 8:58	Agus Arianto
7443	Persetujuan Permintaan	5/2/2017 9:28	Okki Fajrin Dhisati
7443	Penerimaan Barang	5/2/2017 9:28	Ajeng Widya Utami
7422	Persetujuan Permintaan	5/2/2017 9:36	Okki Fajrin Dhisati
7442	Persetujuan Permintaan	5/2/2017 9:49	Okki Fajrin Dhisati
7442	Penerimaan Barang	5/2/2017 9:49	Ajeng Widya Utami
7448	Penerimaan Barang	5/2/2017 9:53	Nugroho Hari Santoso
7448	Persetujuan Permintaan	5/2/2017 9:53	Okki Fajrin Dhisati
7465	Permintaan Barang	5/2/2017 9:57	Anggraeni Permatasari
7409	Persetujuan Permintaan	5/2/2017 10:55	Okki Fajrin Dhisati
7409	Penerimaan Barang	5/2/2017 10:55	Titik Sugiarti
7466	Permintaan Barang	5/2/2017 11:01	Marcha Debby Saraswati
7467	Permintaan Barang	5/2/2017 11:01	Marcha Debby Saraswati
7432	Persetujuan Permintaan	5/2/2017 11:05	Okki Fajrin Dhisati
7432	Penerimaan Barang	5/2/2017 11:05	Titik Sugiarti
7468	Permintaan Barang	5/2/2017 11:28	Rahmatul Fitriyah
7469	Permintaan Barang	5/2/2017 11:32	Nugroho Hari Santoso
7470	Permintaan Barang	5/2/2017 11:48	NURUL FATIMAH
7469	Persetujuan Permintaan	5/2/2017 11:50	Muhammad Fuad Yasin
7469	Penerimaan Barang	5/2/2017 11:50	Nugroho Hari Santoso
7458	Penerimaan Barang	5/2/2017 11:58	Risma Zahra Privea
7458	Persetujuan Permintaan	5/2/2017 11:58	Muhammad Fuad Yasin
7451	Penerimaan Barang	5/2/2017 12:46	Junita Mega Pratiwi
7451	Persetujuan Permintaan	5/2/2017 12:46	Okki Fajrin Dhisati
7453	Persetujuan Permintaan	5/2/2017 12:51	Okki Fajrin Dhisati
.....	.....	.....	.....



*Halaman ini sengaja dikosongkan*

## LAMPIRAN F

Bentuk hasil tampilan prototipe aplikasi *evenl* dalam penelitian ini

Dashboard
Aplikasi
Sistem
Administrator

Home / Aplikasi / Pemetaan / Event Log

PANEL PEMETAAN DATABASE

FORM

Pilih Klien :  
Women Clothing Case ( DBMS : PostgreSQL, DB : women\_clothing, HOST : localhost )

Tampilkan Pemetaan Event Log

Order Penjualan - Konfirmasi Penawaran

Tabel Utama : mail\_tracking\_value  
+ - CASE ID : state\_order\_name  
TIMESTAMP : mail\_tracking\_value.write\_date  
ORIGINATOR : res\_partner\_name

Tabel : mail\_message  
Kolom Tabel : id (integer)  
Berelasi ke : mail\_tracking\_value  
Kolom Relasi : mail\_message\_id (integer)

Tabel : res\_users  
Kolom Tabel : id (integer)  
Berelasi ke : mail\_message  
Kolom Relasi : write\_uid (integer)

Tabel : res\_partner  
Kolom Tabel : id (integer)  
Berelasi ke : res\_users  
Kolom Relasi : partner\_id (integer)

Tabel : sale\_order  
Kolom Tabel : id (integer)  
Berelasi ke : mail\_message  
Kolom Relasi : res\_id (integer)

Kolom : mail\_tracking\_value.new\_value\_char  
Nilai : Sale Order

Kolom : mail\_message.model  
Nilai : sale.order

## Halaman Pemetaan *event logs*

Administrator

Dashboard
Aplikasi
Sistem

Home / Aplikasi / Ekstraksi & Konversi / Bahasa Alami

PANEL BAHASA ALAMI

Pilih Klien :

Women Cloting Case ( DBMS : PostgreSQL, DB : women\_clothing, HOST : localhost )

Bahasa Natural :

tampilkan proses order penjualan

Proses

HASIL EKSTRAKSI BAHASA ALAMI

KETERANGAN EKSTRAKSI BAHASA NATURAL

Proses Bisnis : order penjualan  
Aktivitas : buat penawaran , konfirmasi penawaran , buat tagihan , bayar tagihan , persiapan pengiriman , selesai pengiriman , selesai penjualan  
Bulan : Tidak ditemukan.  
Tahun : Tidak ditemukan.  
Tanggal : Tidak ditemukan.  
Filter Kolom : Tidak ditemukan.  
Urutkan : Tidak ditemukan.

Filter: Type to filter... Q

Show: 10

NO	CASE ID	ACTIVITY/TASK	TIMESTAMP	ORIGINATOR
1	SO001	Buat Penawaran	2018-02-17 15:47:32	Mahendrawathi
2	SO001	Konfirmasi Penawaran	2018-02-17 15:47:38	Mahendrawathi
3	SO001	Selesai Penjualan	2018-02-17 15:47:57	Mahendrawathi
4	SO001	Buat Tagihan	2018-02-17 15:48:21	Mahendrawathi
5	SO001	Bayar Tagihan	2018-02-17 15:48:33	Mahendrawathi
6	SO001	Persiapan Pengiriman	2018-02-17 15:50:09	Mahendrawathi
7	SO001	Selesai Pengiriman	2018-02-17 15:50:09	Mahendrawathi
8	SO002	Buat Penawaran	2018-02-18 05:57:32	Mahendrawathi
9	SO002	Konfirmasi Penawaran	2018-02-18 05:57:43	Mahendrawathi
10	SO002	Selesai Pengiriman	2018-02-18 06:00:55	Mahendrawathi

EXPORT CSV

Showing 1 to 10 of 51 entries

1
2
3
4
5
6

© 2017. Natural Language Interface to a Database (Case event logs extraction, process mining area) by Nambi Sembilu

About
Terms
Contact

## Halaman ekstraksi dengan bahasa alami

*Halaman ini sengaja dikosongkan*





## BIOGRAFI PENULIS

Nambi Sembilu. Lahir di Surabaya, 16 Mei 1990. Merupakan anak ketiga dari 3 bersaudara. Penulis menempuh pendidikan formal 1996-2002 di SD Negeri



Margorejo 1, 2002-2005 di SMP Negeri 13 Surabaya, 2005-2008 SMAN Negeri 17 Surabaya. Tahun 2008 penulis melanjutkan pendidikan strata sarjana di Universitas Airlangga , Fakultas Sains dan Teknologi pada program studi Sistem Informasi. Kemudian pada tahun 2014 memutuskan melanjutkan strata magister dan diterima

sebagai mahasiswa di Institut Teknologi Sepuluh Nopember Surabaya pada program studi Sistem Informasi. Sejak tahun 2011 penulis sudah bekerja sebagai *developer* pada Direktorat Sistem Informasi, Universitas Airlangga. Selain itu penulis juga aktif dalam berbagai macam pengembangan sistem terintegrasi untuk berbagai macam proyek sistem informasi maupun pengembangan profil website untuk perusahaan. Dalam mengisi waktu luang penulis juga sering mengisi kegiatan sosial pada berbagai macam komunitas sosial. Sekarang penulis telah bekerja di PT.Solusi Awan Cerdas Indonesia dengan jabatan *Research and Development Manager* juga sebagai pengajar privat untuk mater-materi yang berkaitan dengan teknologi informasi.

Informasi Email: [nambisembilu@gmail.com](mailto:nambisembilu@gmail.com)